

N° d'ordre : D. U : 2384

EDSPIC :

UNIVERSITE BLAISE PASCAL - CLERMONT II

ECOLE DOCTORALE

SCIENCES POUR L'INGENIEUR DE CLERMONT-FERRAND

T h è s e

Présentée par

YEREMOU TAMTSIA Aurélien

D.E.A en Génie des Procédés

pour obtenir le grade de

DOCTEUR D'UNIVERSITÉ

SPECIALITE : VISION POUR LA ROBOTIQUE

Titre de la thèse

**NOUVELLES CONTRIBUTIONS A L'APPLICATION DES MOMENTS
EN ASSERVISSEMENT VISUEL**

Soutenue publiquement le 11 Octobre 2013 devant le jury :

M. FOGUE Médard - Professeur- Université de Dschang- Bandjoun- Cameroun

M. MEZOUAR Youcef - Professeur - IFMA- Clermont - Ferrand- France

M. TONYE Emmanuel - Professeur- Université de Yaoundé I- Cameroun

M. ELE Pierre - Professeur- Université de Douala- Cameroun

M. TAHRI Omar - Chercheur- Université de Coimbra- Portugal

M. DEMONCEAUX Cédric - Maître de Conférences HDR- Université de Bourgogne-
IUT Le Creusot- France

A ma mère...

Remerciements

Les travaux présentés dans ce manuscrit ont été effectués au LASMEA (Laboratoire des Sciences et Matériaux pour l'Electronique et l'Automatique) et au LETS (Laboratoire d'Electronique et du Traitement du Signal) qui appartient au LSIS (Laboratoire des Systèmes d'Information et de Simulation) respectivement de l'Université Blaise Pascal Clermont Ferrand II et ENSP (Ecole Nationale Supérieure Polytechnique) de l'Université de Yaoundé I.

Je tiens pour cela à remercier les responsables du thème Rosace M. Philippe Martinet et M. Youcef Mezouar, le responsable du LETS de l'ENSP de Yaoundé M. Emmanuel Tonye pour m'avoir accueilli au sein de leur équipe de recherche dans un cadre de travail excellent, et aussi pour leur grande expérience et disponibilité.

Je tiens à remercier :

- M. Haman Djalo co- directeur de ma thèse qui m'a encadré depuis ma première année à l'Université et aussi pour sa rigueur scientifique.

- M. Omar Tahri pour m'avoir guidé tout au long de ce travail, pour son enthousiasme et ses idées toujours plus innovantes, ainsi que ses relectures dans la rédaction d'articles qui m'ont permis de progresser dans mes travaux.

J'exprime ma reconnaissance à tous les membres du jury qui ont consacré leur temps au jugement de ce travail. Il s'agit de :

M. Médard Fogue, directeur de l'IUT FOTSO Victor de Bandjoun, Université de Dschang (Cameroun) , d'avoir accepté de présider ce jury ;

M. Pierre Ele, Professeur à l' ENSP de l'Université de Yaoundé I au Cameroun, et M. Cédric Demonceaux, Professeur à l'Université de Bourgogne en France, d'avoir bien voulu accepter la charge de rapporteur ;

et M. Omar Tahri Chercheur à Institute of Systems Robotics in University of Coimbra (Portugal), d'avoir bien voulu juger ce travail.

Il m'est très difficile de remercier toutes les personnes qui ont contribué de près ou de loin à ce travail. Je veux néanmoins exprimer toute ma gratitude à Roméo Tatsambon Fomena de l'IRISA de Rennes pour ses conseils, toutes l'équipe de recherche du LASMEA pour m'avoir accepté dans cette grande communauté scientifique, la famille Mbang, la famille Chartier, la famille Jackson et la famille Komguep pour leur soutien, Anne Gaëlle Moussongo Ekosso pour sa patience et

ses conseils, le Doyen de la Faculté de Génie Industriel Pr. Robert Nzungwa qui m'a permis de bénéficier la bourse de la formation des formateurs, mes collègues, la famille Fansi, la famille Tchoumbou, la famille Téguia, la famille Andem, la famille Tachiekam, la famille Sop, la famille Fonkoua, la famille Bangoup, sans oublier tous mes amis ...

Un grand Merci à ma famille qui m'a soutenu pendant toutes ces années, et sans qui, rien de tout cela n'aurait été possible.

A mon fils Fessi Yeremou Karel Ivan et ma fille Nkili Yeremou Béliisma Leina, je vous suis très reconnaissant car durant ma formation en France, vous êtes restés très courageux.

Cette thèse n'est guère le fruit d'un travail personnel mais plutôt le fruit de notre collaboration.

Table des matières

Remerciements	i
Table des matières	iii
Table des figures	vii
Liste des Tableaux	ix
Résumé	x
Abstract	xii
Introduction	1
1 Asservissement visuel en robotique : état de l'art et notions de base	5
1.1 Introduction	5
1.2 Notions de géométrie projective	5
1.2.1 Définition des espaces projectifs	6
1.2.2 Homographies	6
1.3 Système de vision	7
1.3.1 Modélisation d'une caméra	8
1.3.2 Les paramètres extrinsèques : Changement de repère ob- jet/caméra	8
1.3.3 Détermination des paramètres intrinsèques	11
1.3.3.1 La projection perspective	12
1.3.3.2 Changement de repère caméra/image	12
1.3.3.3 Modèle complet de la modélisation sans tenir compte des distorsions	13
1.3.3.4 Modèle complet avec distorsion	13
1.3.3.5 Discussion	13
1.4 Asservissement visuel	14
1.4.1 La commande par vision	14

1.4.1.1	Le formalisme des fonctions de tâche	14
1.4.1.2	Type de primitives visuelles	16
1.4.2	Asservissement visuel cinématique	17
1.4.2.1	Principe de la commande par fonction de tâche .	17
1.4.2.2	Asservissement visuel 3D	18
1.4.2.3	Asservissement visuel 2D	21
1.4.2.4	Asservissement visuel hybride	24
1.4.2.5	Discussion	26
1.4.2.6	Conclusion	27
2	Modélisation	29
2.1	Introduction	29
2.2	Mouvement de la caméra et variation des informations visuelles .	29
2.2.1	Primitive Point	30
2.2.2	Les autres primitives géométriques usuelles	31
2.2.2.1	Les moments 2D de l'image	32
2.2.2.2	Définitions et propriétés des moments classiques 2D	32
2.2.2.3	Matrices d'interaction associées aux moments clas- siques 2D :	35
2.2.2.4	Les moments 2D de l'image "shifted"	39
2.2.2.5	Les moments 2D "shifted" pour une image ayant subie une rotation	41
2.2.2.6	Matrices d'interaction associées aux moments 2D "shif- ted"	43
2.2.3	Choix du vecteur d'informations visuelles s	44
2.2.3.1	Informations visuelles pour commander les trans- lations	44
2.2.3.2	Informations visuelles pour commander les rotations	46
2.2.3.3	Nouvelles informations visuelles	48
2.2.3.4	Conclusion	52
3	Etude de la stabilité	53
3.1	Introduction	53
3.2	Stabilité au sens de Lyapunov	54
3.3	Rappels sur la commande référencée capteur des robots manipu- lateurs	55
3.3.1	Approche fonction de tâche appliquée aux asservissements visuels	55
3.3.2	Convergence exponentielle de la fonction de tâche	59
3.3.2.1	Synthèse de la commande.	59
3.3.2.2	Synthèse de commandes dans les différents espaces	61

3.4	Discussion	63
3.5	Matrice d'interaction relative au point P_1	65
3.6	Matrice d'interaction relative au point P_2	68
3.7	Matrice d'interaction relative à l'invariant r_{s3}	70
3.8	Stabilité	75
3.9	Conclusion	75
4	Validation des résultats en asservissement visuel	77
4.1	Introduction	77
4.2	AV2D : Rotation pure	80
4.2.1	Cas de la Baleine	81
4.2.2	Cas de la Feuille de vigne	82
4.2.3	Cas du Rectangle	83
4.3	AV2D : Déplacement générique	84
4.3.1	Cas de la Baleine	84
4.3.2	Cas de la Feuille de vigne	88
4.3.3	Cas du Rectangle	89
4.4	Analyse des résultats et Discussion	91
4.4.1	Analyse des résultats	91
4.4.2	Discussion	94
4.5	Résultats avec des erreurs de modélisation	94
4.6	AV2D : Déplacement générique avec comme paramètres de décalage " a_1 " et " a_2 "	95
4.7	AV2D : Déplacement générique avec comme invariant r_{s3}	95
4.8	Conclusion	97
	Conclusion Générale et Perspectives	105
	Bibliographie	115

Table des figures

1.1	Représentation géométrique des repères utilisés pour la détermination du modèle de la caméra.	9
1.2	Les trois rotations simples.	10
1.3	Asservissement Visuel 3D.	19
1.4	Asservissement Visuel 2D.	22
1.5	Asservissement Visuel hybride.	25
2.1	Projection d'un objet quelconque dans l'image.	30
2.2	Orientation d'un objet et définition des paramètres usuels d'une ellipse.	34
2.3	Représentation de l'image.	40
3.1	cylindre de singularité.	64
4.1	Image utilisée lors de la simulation (a) image d'une baleine (b) image d'une feuille de vigne (c) image d'un rectangle	79
4.2	Image utilisée pour prouver l'invariance (a) image d'une figure symétrique (b) image d'une figure non symétrique	80
4.3	Résultats de Simulation pour un mouvement de rotation pure : image d'une baleine. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".	85
4.4	Résultats de Simulation pour un mouvement de rotation pure : image d'une feuille de vigne. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".	86
4.5	Résultats de Simulation pour un mouvement de rotation pure : image d'un rectangle. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".	87
4.6	Résultats de Simulation pour un mouvement générique : image d'une baleine. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".	91

4.7	Résultats de Simulation pour un mouvement générique : image d'une feuille de vigne. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".	92
4.8	Résultats de Simulation pour un mouvement générique : image d'un rectangle. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".	93
4.9	Résultats de Simulation pour un mouvement générique en présence du bruit : image d'une baleine. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s". . . .	96
4.10	Résultats de Simulation pour un mouvement générique en présence du bruit : image d'une feuille de vigne. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".	97
4.11	Résultats de Simulation pour un mouvement générique en présence du bruit : image d'un rectangle. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".	98
4.12	Résultats de Simulation pour un mouvement générique avec comme paramètres de décalage " a_1 " et " a_2 " : image d'une baleine. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".	99
4.13	Résultats de Simulation pour un mouvement générique avec comme paramètres de décalage " a_1 " et " a_2 " : image d'une feuille de vigne. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".	100
4.14	Résultats de Simulation pour un mouvement générique avec comme paramètres de décalage " a_1 " et " a_2 " : image d'un rectangle. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".	101
4.15	Résultats de Simulation pour un mouvement générique avec comme invariant r_{s3} : image d'une baleine. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".	102
4.16	Résultats de Simulation pour un mouvement générique avec comme invariant r_{s3} : image d'une feuille de vigne. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".	103
4.17	Résultats de Simulation pour un mouvement générique avec comme invariant r_{s3} : image d'un rectangle. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".	104

Liste des tableaux

4.1	Valeurs des Invariants pour un objet de forme symétrique : P_1 comme point de décalage	80
4.2	Valeurs des Invariants pour un objet de forme symétrique : P_2 comme point de décalage	81
4.3	Valeurs des Invariants pour un objet de forme non symétrique : P_1 comme point de décalage	81
4.4	Valeurs des Invariants pour un objet de forme non symétrique : P_2 comme point de décalage	82

Résumé

Cette thèse propose des contributions très prometteuses au sujet du choix des primitives visuelles en asservissement visuel utilisant les moments 2D extraits de l'image. Nous avons proposé une nouvelle manière de résoudre un problème important en asservissement visuel, à savoir la commande du mouvement de rotation suivant les axes orthogonaux à l'axe optique. Ce travail représente une amélioration significative des travaux précédents en asservissement visuel basé sur l'utilisation des moments 2D extraits de l'image pour commander les degrés de liberté des robots manipulateurs. La commande la plus utilisée est connu sous le nom de commande cinématique. L'approche emploie un descripteur global d'image basé sur des moments 2D "shifted" dont les invariants calculés à partir de ces moments 2D "shifted" utilisent des moments d'ordre faible connus pour être robustes au bruit. De plus, ces invariants choisis ne dépendant pas de la forme de l'objet, sont invariants au mouvement de translation, de rotation et d'échelle. Cette nouvelle façon de faire vient ainsi résoudre les problèmes vus dans les travaux précédents relatifs aux choix des combinaisons des moments invariants basés sur les moments centrés et qui dépendent de la forme de l'objet considéré. Dans les travaux précédents, ces invariants sont calculés avec des moments dont l'ordre est compris entre trois et cinq qui sont vulnérables aux bruits de mesure. D'un point de vue asservissement visuel, le travail est basé sur la détermination explicite de la matrice d'interaction calculée à partir des moments 2D "shifted" dont le choix des paramètres de décalages respectent les propriétés d'invariances aux mouvements de translation, de rotation et d'échelle. En procédant ainsi, les informations visuelles choisies montrent la capacité de représenter les objets de formes symétriques et non symétriques. Des résultats de simulations sont présentés pour illustrer la validité de notre proposition.

Mots-clef : moments 2D "shifted", asservissement visuel, Invariants.

Abstract

This thesis proposes nice and very promising contributions about the choice of visual features in image moments-based visual servoing. We proposed a new way towards solving an important problem in Visual Servoing, namely control of non-optic axis rotational degrees of freedom. This work represents a significant improvement respect to previous works on image-based visual servoing of robot manipulators, with the camera in-hand configuration and under the control scheme known as kinematic control. The approach uses global image features that are based on shifted image moments of a planar target. The improvement consists in a particular selection of a combination of shifted image moments of low order such that they do not depend on the planar target shape ; thus solving the problem seen in related previous works where the selection of central or regular image moments combinations depended on the planar target shapes. From a visual servoing point of view, the work is based on the explicit resolution of the interaction matrix related to any shifted image moment, on the appropriate combination of these moments and on the proper selection of the shifted parameters. By doing so, the new features show improved ability to represent symmetrical objects and several kinds of objects defined from closed contours or from a set of points. Six visual features are selected to design a decoupled control scheme when the object is parallel to the image plane. This nice property is then generalized to the case where the desired object position is not parallel to the image plane. Finally simulated results are presented to illustrate the validity of our proposal.

Keywords : Image- moments shifted, visual servoing, Invariants.

Introduction

D'un point de vue méthodologique, l'asservissement visuel consiste à intégrer dans la boucle de commande des robots, des informations extraites des images fournies par des caméras afin de réaliser l'action souhaitée. Les techniques d'asservissement visuel classiques sont efficaces pour une grande classe d'applications mais se heurtent à certaines difficultés [Cha98]. Une difficulté forte limite encore les potentialités des techniques d'asservissement visuelles basées sur l'utilisation de primitives géométriques (points, droites, courbes, ...). En effet, celles-ci nécessitent en général, une mise en correspondance préalable des primitives visuelles (initialisation) et leurs suivis durant toute la phase d'asservissement. Afin de réduire ces contraintes, des travaux ont porté sur l'utilisation de primitives plus globales telles que :

- la transformée de Fourier [CM11]. Dans [CC00] et [CC02], les auteurs dissocient la primitive 2D de sa primitive 3D. Une approximation de la matrice d'interaction dans le cas d'un objet de forme complexe est réalisée. Ce calcul est basé sur la minimisation d'une erreur de modélisation du contour. Cette approche a ensuite été appliquée au cas de la signature polaire et des fonctions paramétriques en portant une attention particulière aux descripteurs de Fourier.
- le signal de luminance [Deg00] et [NNM96]. Dans [CM09a] et [CM09b], une loi de commande dédiée à la luminance a été proposée. Elle s'écrit formellement comme une loi de commande basée sur une méthode de minimisation non linéaire.
- l'intensité lumineuse. Par ailleurs, une approche intéressante, considérant également l'intensité lumineuse, a été proposée dans [KDS⁺07]. Elle est basée sur une méthode de noyaux opérant sur différents signaux permettant d'atteindre un haut degré de découplage de la loi de commande. Seules les translations et la rotation suivant l'axe optique ont été considérées. Nos travaux portent sur la commande des 6 degrés de liberté.

Il s'agit d'utiliser la luminance en tout point de l'image comme informations visuelles dans un schéma d'asservissement visuel. L'originalité de cette approche par rapport aux travaux existants repose sur le calcul analytique de la matrice d'interaction. Pour ce faire, Marchand [Mar07] a considéré le modèle de réflexion de Lambert. Ce qui lui a permis d'obtenir la forme analytique de la matrice d'interaction relative à la luminance.

- les moments 2D extraits de l'image. Les descripteurs de forme ont été largement exploités dans la communauté vision par ordinateur. Ils peuvent généralement être classés comme descripteurs basés contour ou comme descripteurs basés région. Les descripteurs basés contour saisissent seulement l'information du contour de l'objet, tandis que les descripteurs basés région peuvent également capturer la teneur intérieure de la forme et contenir ainsi plus d'information. Parmi des descripteurs basés région, les moments sont certainement les plus populaires. Ils ont été pour la première fois présentés dans la communauté de vision par ordinateur par [Hu62] et ces moments étaient constitués des moments géométriques [Hu62], des moments invariants [Hu62] et des moments orthogonaux [Tea80]. Dans le contexte de l'analyse de forme, les moments sont définis comme projections d'une fonction définissant l'objet sur un ensemble de fonctions caractéristiques des moments considérés. Les moments 2D extraits de l'image sont certainement les descripteurs globaux les plus utilisés en asservissement visuel. Une grande quantité de travaux a été effectuée en asservissement visuel basé sur les moments 2D extraits de l'image. Nous pouvons citer en instance les travaux de [Cha02], [Cha04], [TC05b] et [CTMC04] qui utilisent les moments 2D sur des images de formes complexes. Récemment, les travaux de Mebarki [MKC10] reposent sur l'utilisation des images provenant d'un capteur à ultrasons afin d'intégrer les informations visuelles basées sur des moments 2D extraits de l'image, dans une boucle de commande. Il présente de nouvelles bases théoriques qui ont permis de développer l'expression analytique exacte de la matrice d'interaction qui lie ces informations visuelles à la vitesse de la sonde (ou bien du robot porteur de cette dernière). Pour commander les rotations le long des axes orthogonaux à l'axe optique, il a proposé d'utiliser deux combinaisons des moments invariants où leurs expressions peuvent être trouvées dans [Meb10]. Mais les informations visuelles choisies ne sont pas invariantes vis-à-vis de l'axe optique w_z . L'auteur conclut que le choix de ces invariants est relativement moins robuste aux bruits de mesure dans l'image.

Dans ce travail de thèse, il est question pour nous de choisir une formulation unique d'informations visuelles pour commander le mouvement de rotation le long des axes w_x et w_y orthogonaux à l'axe optique. Nous proposons d'utiliser les moments 2D "shifted", calculé à partir d'un centre de référence autre que le centre de gravité de l'image. Le nouveau centre ainsi choisi, toutes les propriétés d'invariances au mouvement de translation, de rotation et d'échelle sont respectées.

Contributions

Les contributions de cette thèse se situent à trois niveaux :

- le choix des primitives visuelles les plus critiques pour commander le mou-

vement de rotation le long des axes w_x et w_y orthogonaux à l'axe optique, en utilisant des moments 2D "shifted" en lieu et place des moments géométriques. Le centre de gravité de l'objet est décalé en un autre point. Les coordonnées de ce nouveau centre sont choisies de telle sorte que toutes les propriétés d'invariance au translation, rotation et changement d'échelle soient préservées ;

- les primitives visuelles ainsi choisies, ne dépendent pas de la forme de l'objet. Il n'y a aucune restriction en employant ces primitives visuelles sur des objets de formes symétriques ou non. Les images ayant une symétrie le long des axes x et y et symétriques par rapport au centre de gravité de l'objet donnent des valeurs nulles pour les moments centraux d'ordre impairs. Cela est dû à l'emplacement symétrique des pixels de l'image autour du centre de gravité de l'image. Les moments 2D "shifted" produisent des valeurs différentes de zéro pour tous les moments d'ordre pairs et impairs car tous les pixels vont contribuer au calcul des moments dans l'image et résoud ainsi les problèmes causés par les images symétriques ;

- l'utilisation des moments d'ordre faible. Cela permettra de réduire la sensibilité aux bruits de mesure. Les nouveaux moments 2D "shifted" seront plus tolérants aux bruits. Cette façon de faire permet de garantir la stabilité du système.

Organisation du mémoire

Outre cette introduction et une conclusion générale, ce mémoire est constitué de quatre chapitres. Dans le premier chapitre nous fournissons quelques éléments de géométrie projective qui est un outil mathématique bien adapté pour décrire les relations entre la scène tridimensionnelle observée et son image. Cela nous conduira alors au modèle géométrique de la caméra. Nous présentons ensuite un bref état de l'art sur l'asservissement basé sur l'information visuelle en robotique, puis la structure particulière de l'asservissement visuel dans l'image est détaillée.

Dans le second chapitre, nous nous focalisons essentiellement sur le calcul des informations visuelles \mathbf{s} . Afin de développer une méthode reposant sur le lien entre la variation des informations visuelles $\dot{\mathbf{s}}$ et le mouvement de la caméra $\mathbf{T}_{RC/R}$, la stratégie envisagée consiste à exploiter l'historique des mouvements de la caméra et des mesures des indices visuels \mathbf{s} . Dans la suite de ce chapitre, le cas général d'un système à six degrés de liberté est utilisé et $\mathbf{T}_{RC/R}$ est alors défini par : $\mathbf{T}_{RC/R} = (\mathbf{v}, \omega)$ où $\mathbf{v} = (v_x, v_y, v_z)$ et $\omega = (\omega_x, \omega_y, \omega_z)$ sont respectivement les vitesses de translations et de rotation de la caméra. Un état de l'art sur les informations visuelles les plus utilisées est fait dans ce chapitre. Et pour chaque primitive, nous faisons bien de rappeler les primitives visuelles existantes. Par la suite un nouvel ensemble d'informations visuelles est donné en exploitant les moments géométriques 2D non centrés.

Le chapitre trois parlera de la stabilité en asservissement visuel. La notion de stabilité constitue une problématique centrale de la théorie du contrôle. Différents types de problèmes de stabilité peuvent être rencontrés dans l'étude des systèmes dynamiques. Dans cette partie, nous entendons par stabilité des points d'équilibre. La stabilité d'un point d'équilibre est généralement étudiée à l'aide du concept de stabilité au sens de Lyapunov. Par définition, si un système est dans un état d'équilibre, il restera dans cet état pour t variant dans le temps. L'étude de la stabilité au sens de Lyapunov consiste en l'étude des trajectoires du système quand l'état initial est voisin d'un état d'équilibre. Pour résoudre les problèmes associés à des systèmes contrôlés, les mathématiques et l'automatique qui sont des outils bien adaptés à la théorie du contrôle, s'efforcent d'apporter des résultats et des méthodes permettant de comprendre, d'analyser et de résoudre ces problèmes. Les domaines d'application de l'étude de la stabilité sont essentiellement ceux de l'automatique c'est-à-dire la synthèse de lois de commande pour des processus provenant de différents domaines applicatifs comme la robotique. Nous avons à la suite de ce chapitre, proposé la détermination analytique des matrices d'interactions associées aux points \mathbf{P}_1 et \mathbf{P}_2 calculés à partir des nouveaux moments 2D "shifted". Une condition nécessaire et suffisante de stabilité a été donnée.

Enfin, le chapitre quatre est réservé aux résultats. Dans ce chapitre, nous nous intéressons à l'application des travaux de modélisation présentés dans le chapitre II pour des tâches de positionnement par asservissement visuel. Nous présentons les résultats validant le choix des informations visuelles pour l'asservissement visuel par rapport à un objet plan et continu. Dans les séries de simulations que nous présentons dans ce chapitre, nous avons principalement utilisé des objets continus de formes asymétriques. Nous présentons également quelques résultats très satisfaisants obtenus pour un objet symétrique. Quelque soit l'objet utilisé, les mêmes combinaisons d'invariants pour commander les mouvements de rotations le long des axes w_x et w_y orthogonaux à l'axe optique ont été utilisées. En effet, ce choix d'invariants ne dépend pas de l'objet considéré.

Chapitre 1

Asservissement visuel en robotique : état de l'art et notions de base

1.1 Introduction

La première utilisation de la vision en boucle fermée est due à Shirai et Inoue [Shi73] qui décrivent comment un capteur de vision pouvait augmenter la précision du positionnement d'une tâche en robotique. L'asservissement visuel consiste à extraire les informations visuelles issues d'un capteur de vision qui offre la possibilité de travailler avec des quantités importantes d'information. Dans ce chapitre, après avoir rappelé quelques notions de géométrie projective qui est un outil mathématique bien adapté pour décrire les relations entre la scène tridimensionnelle observée et son image, nous décrivons le modèle géométrique d'une caméra. Nous présentons ensuite un bref état de l'art sur l'asservissement basé sur l'information visuelle en robotique, puis la structure particulière de l'asservissement visuel dans l'image est détaillée.

1.2 Notions de géométrie projective

Nous exposons dans ce qui suit quelques notions de base sur la géométrie projective. Cette introduction est construite à partir du chapitre 2 de [Fau93]. La géométrie projective constitue un outil puissant de modélisation en vision par ordinateur et est de ce fait très utilisée dans ce domaine. La caméra est un capteur qui réalise une projection d'un ensemble d'entités géométriques d'une scène tridimensionnelle dans un espace à deux dimensions : le plan image. L'extension de l'espace cartésien \mathbf{R}^n , de dimension n , en l'espace projectif \mathbf{P}^n de dimension $n+1$ permet de manipuler simplement les transformations affines agissant sur \mathbf{R}^n . L'espace projectif \mathbf{P}^n peut donc être vu géométriquement comme une extension

de l'espace euclidien \mathbf{R}^n car il permet la manipulation des entités géométriques à l'infini.

1.2.1 Définition des espaces projectifs

Un point d'un espace projectif \mathbf{P}^n de dimension n , est représenté par un vecteur à $n + 1$ coordonnées $\mathbf{x}_h = [x_1, x_2, x_3, \dots, x_{n+1}]^T$ avec au moins une composante x_i non nulle. Les termes x_i sont appelés coordonnées homogènes du point \mathbf{x}_h . L'espace \mathbf{P}^n est défini par la relation d'équivalence suivante :

Deux vecteurs $\mathbf{x}_h = [x_1, x_2, x_3, \dots, x_{n+1}]$ et $\mathbf{y}_h = [y_1, y_2, y_3, \dots, y_{n+1}]$ représentent le même point de \mathbf{P}^n si et seulement si : $\exists \alpha \neq 0$ tel que

$$\mathbf{x}_h = \alpha \mathbf{y}_h \quad (1.1)$$

Si $x_{n+1} = 0$, alors \mathbf{x}_h est un point à l'infini. Pour $x_{n+1} \neq 0$, un point de l'espace projectif est aussi représenté par :

$$\mathbf{x}_h = [\xi_1, \xi_2, \xi_3, \dots, 1] \quad (1.2)$$

où les coefficients ξ_i sont appelés paramètres projectifs et sont définis par :

$$\xi_i = \frac{x_i}{x_{n+1}} \quad (1.3)$$

1.2.2 Homographies

La transformation linéaire de \mathbf{P}^n dans \mathbf{P}^n définie par une matrice régulière (déterminant non nul) de dimension $(n + 1) \times (n + 1)$ est appelé une homographie. L'ensemble de ses applications forment un groupe appelé le groupe projectif. La matrice associée à une homographie est dénie à un facteur scalaire près non nul :

$$\alpha x = \mathbf{H}y \quad (1.4)$$

où \mathbf{H} est une homographie de l'espace qui conserve le plan à l'infini. Les matrices orthogonales dont le déterminant vaut un sont des matrices de rotation que l'on note \mathbf{R} . Les homographies de la forme :

$$\mathbf{H} = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{T}_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (1.5)$$

sont alors des transformations euclidiennes de l'espace souvent appelées matrice de transformations homogènes. On retrouve ainsi la représentation classique employées en robotique en se plaçant dans un espace cartésien (espace euclidien

muni d'une métrique) de dimension 3. Cet espace est plongé dans l'espace projectif \mathbf{P}^3 (L'espace cartésien 3D dans lequel sont décrites les transformations affines est englobé dans cet espace projectif).

Pour représenter un point \mathbf{P} de l'espace cartésien dont les coordonnées sont exprimées dans un repère \mathbf{R}_i (formé par une base cartésienne $(^iX, ^iY, ^iZ)$ et le point \mathbf{O}_i). Le vecteur $^i\mathbf{P} = [^iP_x, ^iP_y, ^iP_z, 1] \in \mathbb{R}^4$ est employé. Les transformations euclidiennes définissent de façon linéaire les changements de repère. La transformation (translation et/ou rotation) qui amène un repère \mathbf{R}_i en un repère \mathbf{R}_j est décrite par :

$$^i\mathbf{A}_j = \begin{bmatrix} ^i\mathbf{R}_j & ^i\mathbf{T}_j \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (1.6)$$

Où $^i\mathbf{R}_j$ et $^i\mathbf{T}_j$ représentent respectivement la rotation et la translation entre les deux repères \mathbf{R}_i et \mathbf{R}_j .

Ces matrices de transformation homogène permettent d'exprimer, de manière très simple, le changement de coordonnées homogènes des points et des plans dans l'espace cartésien. La situation d'un solide, auquel on a attaché un repère \mathbf{R}_i peut être représentée dans un second repère \mathbf{R}_j par la matrice homogène $^i\mathbf{A}_j$.

1.3 Système de vision

les systèmes de vision (caméras) fournissent un signal video extrêmement riche, prenant la forme d'un tableau numérique de pixels définis par leur position et leur couleur. L'optique géométrique est à l'origine de la formation des images. Plusieurs modèles décrivent la projection opérée par une caméra depuis l'espace 3D observé jusqu'à l'image 2D. Le plus populaire est le modèle sténopé (pinhole), de par sa simplicité et son excellente approximation des caméras perspectives. Dans le modèle sténopé, tous les rayons passent par un seul et même point (le centre optique \mathbf{O}_c), le plan image se situant à une distance f (la distance focale) derrière le centre optique. Par convention, l'axe optique est porté par l'axe z_c , définissant ainsi le plan image. La projection dans l'image peut se décomposer en trois transformations : un changement de repère de l'objet à la caméra, une projection linéaire de centre \mathbf{O}_c (origine du repère \mathbf{R}_c attaché à la caméra) de \mathbf{P}^3 dans \mathbf{P}^2 et d'un changement de repère dans \mathbf{P}^2 . La connaissance des paramètres intrinsèques est nécessaire pour pouvoir interpréter les mesures pixelliques de la caméra en termes géométriques, et ainsi, faire le lien entre l'image et l'environnement. L'opération consistant à identifier ces paramètres est la calibration de caméra. Dans ce qui suit, nous définissons d'abord l'ensemble des paramètres du modèle d'une caméra. Ensuite, nous estimons les paramètres du modèles.

1.3.1 Modélisation d'une caméra

Les paramètres intrinsèques et extrinsèques permettent de modéliser une caméra. Onze paramètres sont nécessaires pour la modélisation.

paramètres intrinsèques

Les paramètres intrinsèques qui modélisent la géométrie interne et les caractéristiques optiques du capteur, sont au nombre de cinq. Le modèle de caméra représenté sur la Figure 1.1 relie les coordonnées pixeliques aux coordonnées métriques. Il dépend :

- des coordonnées pixeliques (u_0, v_0) représentant les coordonnées (en pixel) dans l'image, de l'intersection de l'axe optique de la caméra et du plan image. Ce point s'appelle également le centre optique \mathbf{O}_c ;
- les paramètres k_u et k_v qui sont les facteurs d'échelle exprimés en pixel par mètre. Ces paramètres sont les facteurs verticaux et horizontaux qui définissent la taille d'un pixel ;
- la distance focale f de la caméra c'est-à-dire la distance entre le centre de projection (plan image) et \mathbf{O}_c le centre de l'image.

paramètres extrinsèques

Six paramètres extrinsèques sont nécessaires pour modéliser la relation qui lie le repère associé à la caméra au repère de la scène où se trouve les objets. L'ensemble des paramètres extrinsèques sont les paramètres liés à la position et à l'orientation du repère de la caméra en fonction du repère de la scène dans l'espace :

- les trois rotations autour des trois axes déterminées par une matrice de rotation 3x3 nommée ${}^c\mathbf{R}_w$;
- Les trois translations déterminées par un vecteur translation 3x1 nommé ${}^c\mathbf{T}_w$.

1.3.2 Les paramètres extrinsèques : Changement de repère objet/caméra

Cette première transformation permet d'obtenir les coordonnées des points de l'objet observé, exprimées dans un repère \mathbf{R}_m (encore appelé repère monde) attaché à cet objet dans le repère de la caméra \mathbf{R}_c . Le point \mathbf{P} a pour coordonnées $({}^mP_x, {}^mP_y, {}^mP_z)^T$ dans le repère \mathbf{R}_m et $({}^cP_x, {}^cP_y, {}^cP_z)^T$ dans le repère \mathbf{R}_c . La relation géométrique qui existe entre ces deux repères permet de définir les paramètres extrinsèques du modèle, de sorte que ces paramètres déterminent la position et l'orientation du repère de la caméra par rapport au repère du monde.

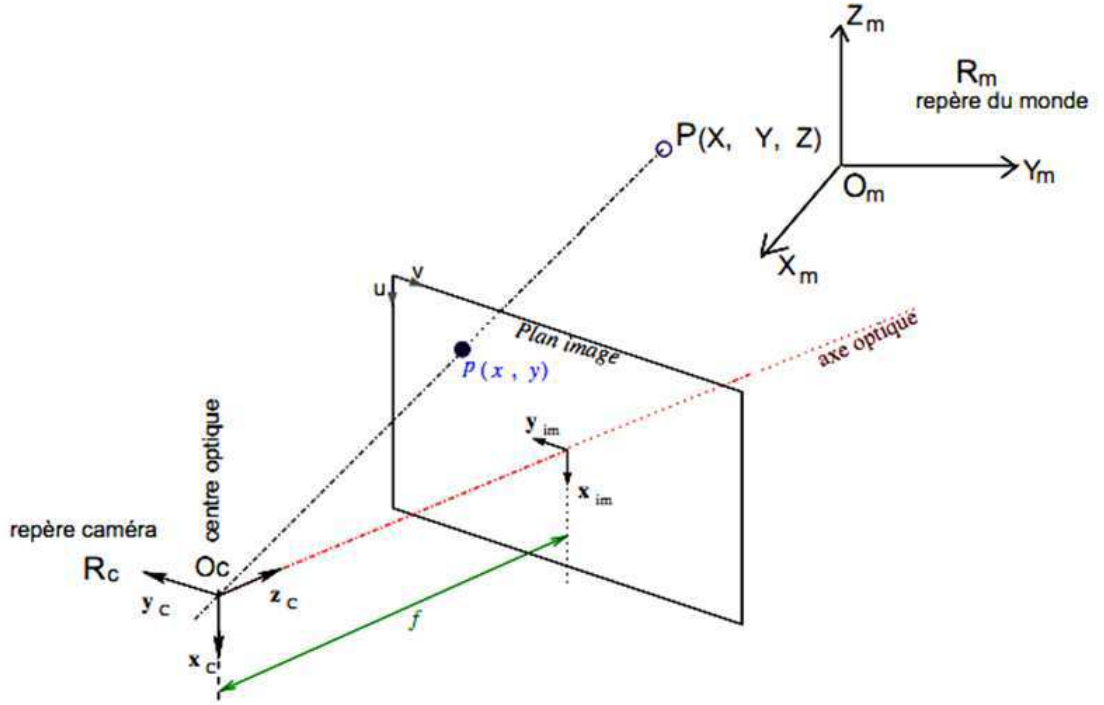


FIGURE 1.1 – Représentation géométrique des repères utilisés pour la détermination du modèle de la caméra.

Pour déterminer la position et l'orientation d'un objet, considérons deux repères \mathbf{R}_m (repère monde) et \mathbf{R}_c (repère caméra) dont les axes respectifs sont (X_m, Y_m, Z_m) et (x_c, y_c, z_c) . La pose de l'objet sera décrite par le mouvement entre les deux repères. La translation correspond simplement au vecteur ${}^c\mathbf{T}_m$ défini par :

$${}^c\mathbf{T}_m = \mathbf{O}_c - \mathbf{O}_m \quad (1.7)$$

Une rotation θ_X autour de l'axe X_m (voir Figure 1.2) s'écrit de la manière suivante :

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_X & -\sin \theta_X \\ 0 & \sin \theta_X & \cos \theta_X \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix} \quad (1.8)$$

qui s'écrit encore sous la forme :

$$P_c = \mathbf{R}(\theta_X) P_m$$

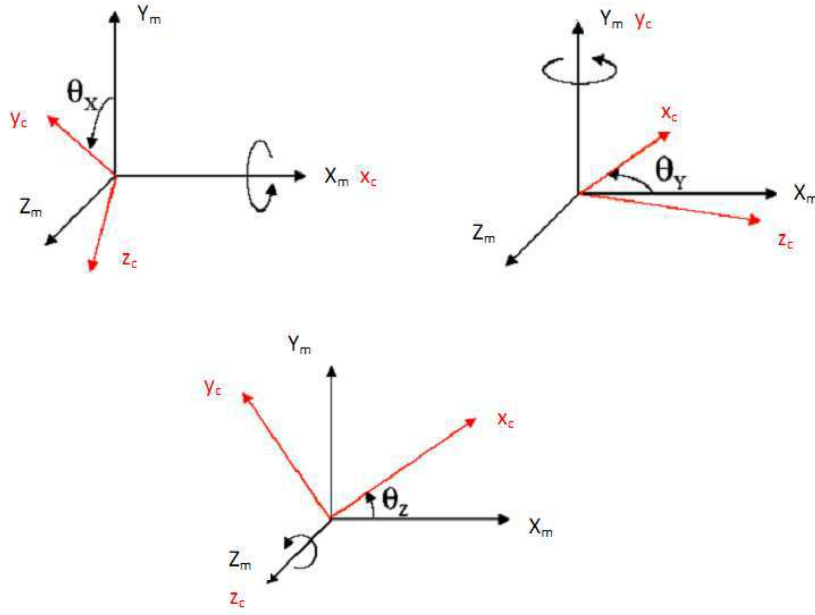


FIGURE 1.2 – Les trois rotations simples.

avec :

$$\mathbf{R}(\theta_X) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_X & -\sin \theta_X \\ 0 & \sin \theta_X & \cos \theta_X \end{bmatrix} \quad (1.9)$$

En procédant de la même façon, on obtient suivant l'axe Y_m et l'axe Z_m les relations suivantes :

$$P_c = \mathbf{R}(\theta_Y) P_m$$

avec

$$\mathbf{R}(\theta_Y) = \begin{bmatrix} \cos \theta_Y & 0 & \sin \theta_Y \\ 0 & 1 & 0 \\ -\sin \theta_Y & 0 & \cos \theta_Y \end{bmatrix} \quad (1.10)$$

$$P_c = \mathbf{R}(\theta_Z) P_m$$

avec :

$$\mathbf{R}(\theta_Z) = \begin{bmatrix} \cos \theta_Z & -\sin \theta_Z & 0 \\ \sin \theta_Z & \cos \theta_Z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.11)$$

La matrice de rotation décrivant l'orientation du repère \mathbf{R}_c par rapport au repère \mathbf{R}_m est donnée par la relation suivante :

$${}^c\mathbf{R}_m = \mathbf{R}(\theta_X) \mathbf{R}(\theta_Y) \mathbf{R}(\theta_Z) \quad (1.12)$$

Pour un point \mathbf{P}_m appartenant au repère \mathbf{R}_m et un point \mathbf{P}_c appartenant au repère \mathbf{R}_c ; la relation qui existe entre les deux points est la suivante :

$$\mathbf{P}_c = {}^c\mathbf{R}_m \mathbf{P}_m \quad (1.13)$$

En résumé, les points 3D projetés dans le repère de la caméra sont donnés par la relation entre les deux repères :

$$\begin{bmatrix} {}^c x \\ {}^c y \\ {}^c z \end{bmatrix} = {}^c\mathbf{R}_m \cdot \begin{bmatrix} {}^m X \\ {}^m Y \\ {}^m Z \end{bmatrix} + {}^c\mathbf{T}_m \quad (1.14)$$

Pour exprimer les coordonnées $({}^m P_x, {}^m P_y, {}^m P_z)^T$ du point \mathbf{P} , appartenant au repère du monde dans le repère lié à la caméra, nous définissons ses nouvelles coordonnées $({}^c P_x, {}^c P_y, {}^c P_z)^T$ de la manière suivante :

$$\begin{bmatrix} {}^c P_x \\ {}^c P_y \\ {}^c P_z \end{bmatrix} = {}^c\mathbf{R}_m \cdot \begin{bmatrix} {}^m P_x \\ {}^m P_y \\ {}^m P_z \end{bmatrix} + {}^c\mathbf{T}_m \quad (1.15)$$

que nous pouvons réécrire comme simple matrice en représentation homogène de la manière suivante :

$$\begin{bmatrix} {}^c P_x \\ {}^c P_y \\ {}^c P_z \\ 1 \end{bmatrix} = \begin{bmatrix} {}^c\mathbf{R}_m & {}^c\mathbf{T}_m \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^m P_x \\ {}^m P_y \\ {}^m P_z \\ 1 \end{bmatrix} \quad (1.16)$$

Une forme simplifiée est :

$$\begin{bmatrix} {}^c P \\ 1 \end{bmatrix} = {}^c\mathbf{M}_m \cdot \begin{bmatrix} {}^m P \\ 1 \end{bmatrix} \quad (1.17)$$

avec

$${}^c\mathbf{M}_m = \begin{bmatrix} {}^c\mathbf{R}_m & {}^c\mathbf{T}_m \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (1.18)$$

La matrice ${}^c\mathbf{M}_m$ représente les paramètres extrinsèques. Son inverse est donné par :

$${}^c\mathbf{M}_m^{-1} = \begin{bmatrix} {}^c\mathbf{R}_m^t & -{}^c\mathbf{R}_m^t {}^c\mathbf{T}_m \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (1.19)$$

1.3.3 Détermination des paramètres intrinsèques

Jusqu'à présent, nous nous sommes contentés de déterminer les coordonnées d'un point \mathbf{P} appartenant au repère du monde, dans le repère de la caméra. A présent, déterminons les coordonnées de ce point dans le repère \mathbf{R}_{im} du plan image. En d'autre terme, exprimons la relation entre un point 3D et son projeté 2D dans le plan image.

1.3.3.1 La projection perspective

Le modèle sténopé de la figure 1.1 est la plus utilisée en projection perspective. Ce modèle correspond à la projection perspective d'un point \mathbf{P} en un point image \mathbf{p} de coordonnées homogènes $p = [u, v, 1]^T$ exprimées en pixels. Le changement de variables pour passer des coordonnées $[u, v, 1]^T$ d'un point \mathbf{p} exprimés en pixels, aux coordonnées $[x, y, 1]^T$ de ce même point mais exprimé en mètre est donné par :

$$\begin{cases} u = k_u x + u_0 \\ v = k_v y + v_0 \end{cases} \quad (1.20)$$

(u_0, v_0) représente les coordonnées (en pixel) dans l'image, de l'intersection de l'axe optique de la caméra et du plan image. Ce point s'appelle, également, le centre optique, k_u et k_v sont les facteurs d'échelle exprimés en pixel par mètre. Cette relation peut encore s'écrire sous la forme :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1.21)$$

la matrice des paramètres intrinsèques de la caméra est donc donnée par :

$$\mathbf{K} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.22)$$

1.3.3.2 Changement de repère caméra/image

Les coordonnées $[x, y, 1]^T$ d'un point \mathbf{p} du plan image sont liées aux coordonnées $({}^cP_x, {}^cP_y, {}^cP_z)^T$ d'un point qui est le projeté d'un point \mathbf{P} appartenant au repère monde par la relation :

$$\begin{cases} x = f \frac{{}^cP_x}{{}^cP_z} \\ y = f \frac{{}^cP_y}{{}^cP_z} \end{cases} \quad (1.23)$$

En coordonnées homogènes, on peut représenter l'équation (1.23) par une multiplication matrice-vecteur de la forme :

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} {}^cP_x \\ {}^cP_y \\ {}^cP_z \\ 1 \end{pmatrix} \quad (1.24)$$

Nous combinons maintenant les équations (1.21) et (1.24) pour modéliser la projection complète du repère caméra vers le repère pixels :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fk_u & 0 & u_0 & 0 \\ 0 & fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^cP_x \\ {}^cP_y \\ {}^cP_z \\ 1 \end{bmatrix} \quad (1.25)$$

La calibration d'une caméra revient à modéliser celle-ci c'est à dire trouver ses paramètres intrinsèques [Tsa87]. Si un des paramètres du modèle devrait changer, il faudra recalibrer toute la caméra [Bey92].

1.3.3.3 Modèle complet de la modélisation sans tenir compte des distorsions

Pour obtenir le modèle algébrique complet de la caméra, nous combinons simplement les équations (1.16) et (1.25) :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fk_u & 0 & u_0 & 0 \\ 0 & fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^cR_m & {}^cT_m \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^mP_x \\ {}^mP_y \\ {}^mP_z \\ 1 \end{bmatrix} \quad (1.26)$$

1.3.3.4 Modèle complet avec distorsion

Le modèle géométrique de la caméra ainsi défini est un modèle simplifié et idéal. Il est largement suffisant dans la plupart des applications de vision. Cependant, lorsqu'une précision élevée est nécessaire, un modèle plus complexe doit être employé. Les lentilles des caméras, par leur symétrie sphérique, provoquent une distorsion, c'est-à-dire une déformation de l'image réelle. On décompose la distorsion en une composante radiale et une composante tangentielle. La première est due à l'asymétrie des lentilles, la deuxième au mauvais alignement des lentilles. Il est possible de corriger l'image captée si l'on possède les paramètres liés à la caméra (les paramètres intrinsèques), on peut alors reformer l'image par interpolation des pixels ayant subi préalablement un déplacement inverse à celui de la distorsion. En général, le modèle sténopé est complété par des termes correctifs caractérisant la déformation de l'image. Pour des modèles de caméra avec distorsion, on pourra consulter [ML95],[Ré98] et [Zha96].

1.3.3.5 Discussion

Le modèle utilisé jusqu'ici est linéaire. En réalité, la lumière ne voyage pas toujours en ligne droite dans un système optique. En traitement d'image, l'opération de calibration de caméra revient à modéliser le processus de formation des

images, c'est-à-dire trouver la relation entre les coordonnées spatiales d'un point de l'espace avec le point associé dans l'image prise par la caméra. Pour calibrer une caméra, le lecteur intéressé pourra par exemple se référer à [Bou94] , [RH93] et [Fau93].

1.4 Asservissement visuel

Nous allons aborder dans cette partie les bases de la commande par vision vue sous l'angle des performances et de la robustesse. L'expression "asservissement visuel" (en anglais, "visual servoing") a été introduite par Agin [Agi77] pour décrire un bras manipulateur contrôlé à partir des informations extraites des images fournies par un capteur de vision. Une étape fondamentale dans cette approche est le choix des primitives qui seront utilisées. Ces informations visuelles sont l'équivalent vectoriel du signal délivré par un capteur de vision classique. Les schémas d'asservissement visuel peuvent être divisés en deux principales approches selon le choix des informations visuelles utilisées. La première est l'asservissement visuel 3D en position (ou PBVS pour position-based visual servoing). La deuxième est l'asservissement visuel 2D dans image (ou IBVS pour image-based visual servoing). Pour pallier aux inconvénients de ces deux méthodes, une approche hybride a été envisagée. Dans la suite de ce paragraphe, un état de l'art est fait sur les bases théoriques concernant l'asservissement visuel.

1.4.1 La commande par vision

Déplacer un système robotique en fonction des informations visuelles extraites des images issues d'un capteur de vision est l'objectif principal de la commande par vision. Une tâche robotique (positionnement) consiste à amener le robot d'une position initiale quelconque dans une position désirée définie par une image associée à cette position. Si le vecteur d'informations visuelles est correctement choisis, il existe une unique position de la caméra qui assure la régulation à zéro de l'erreur entre les informations visuelles désirées et leurs valeurs courantes. Il existe différentes manières de construire une commande référencée vision en fonction de l'architecture de commande, de l'espace dans lequel les primitives sont définies et de la configuration de la caméra [LW87] et [PKA93].

1.4.1.1 Le formalisme des fonctions de tâche

Le formalisme de la fonction de tâche [CS91] définit une erreur \mathbf{e} comme l'écart entre la valeur courante et la valeur désirée pour les variables définissant la tâche de positionnement. L'erreur \mathbf{e} dépend de la position du robot, du temps et éventuellement d'autres grandeurs liées à l'environnement. L'objectif de ce

type de commande est de minimiser un critère d'erreur, appelé la fonction de tâche [CS91] et [BE92], qui dépend du signal de régulation $\mathbf{s}(\mathbf{r}(t), t)$ et de la configuration désirée \mathbf{s}^* . La fonction de tâche est généralement définie comme suit :

$$\mathbf{e}(\mathbf{r}(t), t) = \mathbf{C}(\mathbf{s}(\mathbf{r}(t), t) - \mathbf{s}^*) \quad (1.27)$$

où \mathbf{C} est une matrice dite de combinaison qui permet de tenir compte d'une éventuelle redondance d'informations, $\mathbf{s}(\mathbf{r}(t), t)$ est un vecteur contenant les valeurs courantes des informations capteurs choisies. Ce vecteur exprime le fait que les données visuelles à l'instant courant dépendent non seulement de la situation de la caméra \mathbf{r} par rapport à un repère de la scène, mais également du temps.

La fonction de tâche ainsi définie apparaît comme une fonction d'erreur et le problème de la régulation de cette fonction de tâche \mathbf{e} consiste à amener cette erreur à zéro et à l'y maintenir. Sur cette base, la convergence de la fonction de tâche vers zéro garantit l'exécution du mouvement de la caméra vers la position désirée. Par conséquent, afin d'imposer une décroissance exponentielle de la fonction de tâche \mathbf{e} , il suffit de calculer le vecteur de commande qui représente la vitesse à envoyer aux actionneurs du robot pour accomplir une tâche. La loi de commande la plus utilisée, est celle de la décroissance exponentielle découplée de la fonction de tâche. Elle s'exprime sous la forme :

$$\dot{\mathbf{e}}(\mathbf{r}(t), t) = \mathbf{C}\dot{\mathbf{s}}(\mathbf{r}(t), t) - \mathbf{C}\mathbf{L}_s\mathbf{T}_{RC/R} + \frac{\partial \mathbf{e}}{\partial t} \quad (1.28)$$

Où $\frac{\partial \mathbf{e}}{\partial t} = \mathbf{C}\frac{\partial \mathbf{s}}{\partial t}$ représente les variations de \mathbf{e} dues au mouvement propre de la cible visuelle. Pour contrôler la situation complète de la caméra, il est nécessaire de disposer d'au moins six informations visuelles indépendantes ($k = 6$) pour pouvoir contrôler les six degrés de liberté du système (trois degrés de liberté pour la position et trois degrés de liberté pour l'orientation). Dans ce cas, la matrice d'interaction \mathbf{L}_s sera de rang plein. Comme le système commandé est du premier ordre (commande en vitesse), la réponse généralement souhaitée est une décroissance exponentielle de l'erreur. Une loi de commande classiquement utilisée et garantissant la décroissance exponentielle de la fonction de tâche \mathbf{e} est la suivante :

$$\mathbf{T}_{RC/R} = (\mathbf{C}\mathbf{L}_s)^{-1} \left(-\lambda \mathbf{e} - \frac{\partial \mathbf{e}}{\partial t} \right) \quad (1.29)$$

\mathbf{L}_s est la matrice d'interaction associée aux informations visuelles \mathbf{s} . Cette matrice relie la vitesse dans l'espace aux variations du signal \mathbf{s} extrait de l'image. Par ailleurs, la matrice de combinaison est généralement choisie de façon à ce que sa variation dans le temps soit nulle (constante) ou négligeable devant la variation du signal de régulation \mathbf{s} .

λ est le facteur de réglage de la vitesse de décroissance de la fonction de tâche \mathbf{e} .

$\mathbf{T}_{RC/R}$ est le torseur cinématique. Il représente la vitesse de la caméra par rapport à l'objet et s'écrit $\mathbf{T}_{RC/R} = (\mathbf{v}, \omega)$ où $\mathbf{v} = (v_x, v_y, v_z)$ et $\omega = (\omega_x, \omega_y, \omega_z)$ sont respectivement les vitesses de translations et de rotation du mouvement de la caméra.

Pour un meilleur comportement de la matrice d'interaction \mathbf{L}_s , la matrice de combinaison \mathbf{C} peut être choisie comme étant la pseudo-inverse de la matrice d'interaction. On a alors $\mathbf{C} = \mathbf{L}_s^+$ où \mathbf{L}_s^+ représente la pseudo-inverse de la matrice d'interaction \mathbf{L}_s . Ce choix permet de linéariser et de découpler le système commandé. Cependant, il n'est pas possible en pratique d'obtenir la valeur exacte de la matrice d'interaction, c'est pourquoi le vecteur de commande est calculé sur la base d'une estimation de cette matrice $\widehat{\mathbf{L}}_s^+$.

Comme le facteur d'amortissement est choisi positif, la condition de stabilité locale de la commande est remplie si le produit de la matrice d'interaction réelle par la matrice estimée est défini positif [CS91] :

$$\widehat{\mathbf{L}}_s^+ \mathbf{L}_s > 0 \quad (1.30)$$

L'équation (1.28) montre que l'erreur de suivi n'est nulle que si la dérivée partielle de la fonction de tâche par rapport au temps ainsi que la matrice d'interaction sont parfaitement connues.

1.4.1.2 Type de primitives visuelles

Il existe un grand nombre d'informations visuelles, encore appeler primitives, qui peuvent être mises à profit dans une tâche d'asservissement visuel. l'extraction des primitives visuelles réduit ainsi l'image à un vecteur de paramètres. Cette étape est souvent coûteuse en temps de calculs et oblige à contourner le problème en utilisant des images binaires (objet Blanc sur fond noir) randant ainsi les conditions expérimentales parfaites. Une primitive visuelle est une forme géométrique élémentaire (résultat de la projection dans l'image d'une forme géométrique 3D) que l'on peut caractériser sous forme d'un vecteur de paramètres selon la représentation choisie. Une bonne primitive visuelle doit pouvoir être extraite facilement et sans ambiguïté d'une image quelle que soit la vue de la scène considérée. Les primitives que nous retrouvons le plus souvent, sont des primitives géométriques de types, points [BE92], droites [BE92] ou [NAH02], cercles [BE92] ou [MC02b], sphères [TFC09]. Les paramètres qui permettent de décrire ces différentes primitives peuvent être des paramètres de position définis dans l'image ou dans l'espace en utilisant une représentation de type cartésienne ou polaire [IO05] mais très souvent aussi cylindrique [IO05]. Afin d'assurer le découplage et de linéarité [CH01], il est même possible de combiner plusieurs paramètres où chacun

est utilisé pour la commande d'un degré de liberté. Des primitives plus complexes ont aussi été évaluées dans la littérature : Dans [Cha04], [TC05b], [KI07], les moments d'un objet complexe sont considérés pour réaliser l'asservissement visuel. Les primitives visuelles ne sont pas toujours purement géométriques. En effet, des paramètres cinématiques obtenus à partir de la dérivée des paramètres géométriques peuvent être utilisés pour la commande [PMG96], [CC01b], [LD08] et [MC10]. Dans [PMG96], les paramètres cinématiques sont associés à des paramètres géométriques afin d'améliorer les performances de la commande par vision alors que dans [CC01b], [CC01a], seuls les paramètres cinématiques sont utilisés afin d'améliorer la robustesse de la commande dans le cas où les images sont trop variables et bruitées pour suivre des primitives tout au long de la commande. De façon plus globale, ces primitives peuvent être considérées comme étant des primitives spatio-temporelles [SJ07]. Néanmoins, les objets considérés ou les scènes observées sont souvent représentés par un ensemble de points puisqu'ils facilitent le traitement d'image. Il faut souligner que le comportement de la structure de commande dépend fortement du type de primitives utilisées [Cha98], [Has98] et de leur nombre [Mic93]. La position désirée dans l'image est généralement définie à l'initialisation par apprentissage, ce qui permet d'éviter d'utiliser le modèle de la caméra pour générer l'image désirée.

1.4.2 Asservissement visuel cinématique

Pour contrôler un robot avec un retour visuel, les lois de commandes les plus répandues sont purement cinématiques. La commande cinématique consiste à envoyer périodiquement des consignes de vitesses articulaires au contrôleur bas niveau du robot. La cadence de rafraîchissement des consignes est imposée entre autres par la fréquence d'acquisition des images, la durée des traitements nécessaires à l'extraction de l'information visuelle et le calcul de la loi de commande. Les schémas de commandes cinématiques sont basés sur le formalisme de la régulation à zéro de la fonction de tâche sur un horizon temporel $[0, T]$. Dans notre cas, cette fonction représentera l'erreur d'asservissement \mathbf{e} .

1.4.2.1 Principe de la commande par fonction de tâche

Généralement, une tâche robotique peut être exprimée comme la régulation, sur un certain horizon temporel, d'une fonction $\mathbf{e}(\mathbf{q}, t)$ de classe C^2 appelée fonction de tâche (où \mathbf{q} est le vecteur de coordonnées articulaires du robot). Les fonctions de tâches classiques sont :

- $\mathbf{e}(\mathbf{q}, t) = \mathbf{q} - \mathbf{q}^*(t)$ où $\mathbf{q}^*(t)$ est une trajectoire désirée dans l'espace articulaire.

- $\mathbf{e}(\mathbf{q}, t) = \mathbf{r}(\mathbf{q}) - \mathbf{r}^*(t)$ où $\mathbf{r}^*(t)$ est une trajectoire désirée dans l'espace cartésien.

- $\mathbf{e}(\mathbf{q}, t) = \mathbf{e}(\mathbf{s}(\mathbf{q}) - \mathbf{s}^*(t))$, où $\mathbf{s}^*(t)$ est une trajectoire désirée dans l'espace du capteur.

En introduisant cette nouvelle notion de fonction de tâche, il est clair que la deuxième classification peut être faite sur la base du choix de cette dernière. En effet, on peut par exemple construire cette fonction de la manière suivante :

- $\mathbf{e}(\mathbf{q}, t) = \mathbf{r}(\mathbf{s}(\mathbf{q})) - \mathbf{r}(\mathbf{s}^*(t))$. La position \mathbf{r} de l'effecteur du robot est estimée à partir des informations visuelles. Cette approche est appelée Asservissement visuel 3D car le contrôle de la caméra se fait dans l'espace cartésien [EBC04].

- $\mathbf{e}(\mathbf{q}, t) = \mathbf{C}(\mathbf{s}(\mathbf{q}) - \mathbf{s}^*(t))$ où \mathbf{C} est une matrice qui permet de tenir compte d'une éventuelle redondance d'informations. La fonction de tâche est construite à partir de la différence entre les informations courantes et désirées. Cette approche est appelée Asservissement visuel 2D car le contrôle se fait dans l'image.

Ces deux méthodes ne sont pas les seules existantes, mais elles sont les plus répandues dans la littérature. L'asservissement visuel 2D1/2 est une autre méthode qu'on citera dans ce travail qui est en réalité, une dérivée des deux méthodes précédentes [MCB99].

1.4.2.2 Asservissement visuel 3D

On parle d'asservissement visuel 3D (PBVS pour Position-Based Visual Servoing) lorsque le contrôle de déplacement d'un robot se fait dans l'espace cartésien (confère figure 1.3). La grandeur à asservir \mathbf{r} est une mesure tridimensionnelle. Elle peut représenter des coordonnées de points 3D de l'objet observé ou directement caractériser la pose relative de l'effecteur du robot par rapport à un repère cible à atteindre. Dans tous les cas il est nécessaire de reconstruire cette pose à partir d'informations visuelles extraites de l'image et d'un modèle géométrique de l'objet d'intérêt. Un asservissement visuel 3D nécessite donc une étape de reconstruction permettant de fournir une mesure d'attitude de l'objet. Les techniques de reconstruction de pose utilisent pour la plupart des cas des primitives géométriques extraites de l'image ainsi qu'une connaissance du modèle 3D de l'objet observé. Le signal de référence peut représenter soit la pose relative de la caméra par rapport à la cible (asservissement visuel 3D pose) soit les coordonnées 3D des primitives observées (asservissement visuel 3D point par exemple). Si la primitive choisie dans le cas général est un point, alors la matrice d'interaction relative à cette primitive \mathbf{L}_{3D} est donnée par l'équation (1.33). Le passage de l'espace capteur à l'espace cartésien passe forcément par une reconstruction 3D, qui consiste en général à retrouver l'information perdue soit à partir d'une connaissance a priori de la cible (modèle 3D de l'objet) ou alors à partir d'une mesure redondante des primitives en question grâce à plusieurs caméras où une caméra qui se

déplace [HZ00]. En réalité, l'asservissement visuel 3D , consiste à s'intéresser aux choix du signal de régulation et à la manière de l'exploiter.

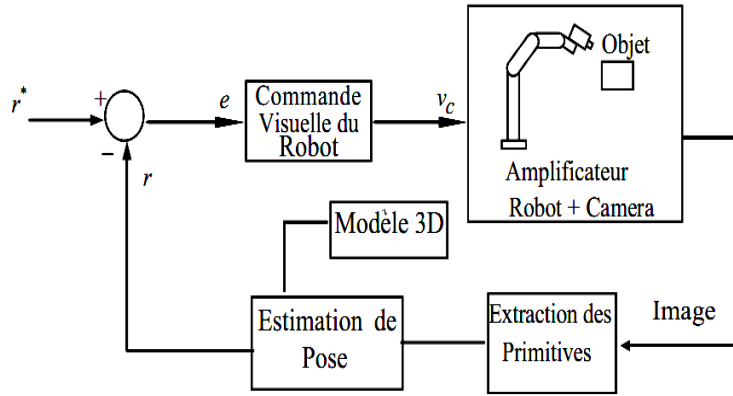


FIGURE 1.3 – Asservissement Visuel 3D.

Pour décrire la pose de la caméra aussi bien en ce qui concerne la position (coordonnées cartésiennes, sphériques, cylindriques, etc.), qu'en ce qui concerne l'orientation (angles d'Euler, quaternions, etc.), différents choix de paramétrages peuvent être utilisés. Sachant que le choix du signal se fait par rapport aux propriétés de découplage de la commande, qui dépend de la structure cinématique du robot, le choix du signal de référence devrait aussi tenir compte de la structure du robot. Cependant, pour un robot à 6 degrés de liberté, le vecteur de commande est généralement choisi comme étant le torseur cinématique, qui peut être ensuite traduit en vitesses articulaires en utilisant la jacobienne cinématique inverse du robot. La synthèse de commande en asservissement visuel 3D consiste donc à choisir les paramètres de pose qui présentent les meilleures propriétés de découplage par rapport au torseur cinématique et de calculer la matrice d'interaction qui lui est associée. Cette matrice d'interaction obtenue doit être définie et inversible dans tout l'espace de travail de notre robot. Ceci impose donc une forte contrainte liée au choix des paramètres de la pose. Sachant que le vecteur de commande est le torseur cinématique, alors les coordonnées cartésiennes sont les paramètres intéressants en ce qui concerne les paramètres de position. La dérivée temporelle de ce vecteur n'est autre que la vitesse de translation du vecteur de commande. Pour exprimer la matrice d'interaction relatives à ces paramètres, il suffit d'exprimer la vitesse de l'origine de l'objet par rapport à la vitesse de la caméra exprimée dans le repère de la caméra. On obtient alors :

$${}^c\dot{\mathbf{P}}(t) = -{}^c\mathbf{v} - {}^c\boldsymbol{\omega} \times {}^c\mathbf{OP} \quad (1.31)$$

La forme matricielle de cette équation est :

$${}^c\dot{\mathbf{P}}(t) = \begin{bmatrix} -\mathbf{I}, & [{}^c\mathbf{OP}]_{\times} \end{bmatrix} \begin{bmatrix} {}^c\mathbf{v} \\ {}^c\omega \end{bmatrix} = \mathbf{L}_{3D} \mathbf{T}_{RC/R} \quad (1.32)$$

avec :

$$\mathbf{L}_{3D} = \begin{bmatrix} -\mathbf{I}, & [{}^c\mathbf{OP}]_{\times} \end{bmatrix} \quad (1.33)$$

où \mathbf{L}_{3D} représente la matrice d'interaction du point 3D

Afin de contrôler l'orientation de la caméra, le vecteur de paramètres qui semble être le plus approprié est le vecteur $\theta \mathbf{u}$ [CS91, Mal98], où θ et $\mathbf{u} = [u_x, u_y, u_z]^T$ représentent respectivement l'angle et l'axe de rotation. Ce vecteur représente l'orientation de la caméra entre le repère courant et le repère désiré. En plus du fait qu'il soit à représentation minimale (3 éléments correspondant à 3 d.d.l), et que sa jacobienne par rapport à la vitesse de rotation est inversible dans tout l'espace d'orientation, ce vecteur est le vecteur propre réel de la matrice d'orientation homogène qui lui correspond et il présente des propriétés intéressantes par rapport à cette dernière. La dérivée de $\theta \mathbf{u}$ peut s'écrire en fonction de la vitesse de rotation ω , composante du torseur cinématique de la caméra :

$$\frac{d\theta \mathbf{u}}{dt} = \mathbf{L}_{\omega} \omega \quad (1.34)$$

où l'expression de \mathbf{L}_{ω} est donnée dans [Mal98] par :

$$\mathbf{L}_{\omega}(\mathbf{u}, \theta) = \mathbf{L}_{\omega} = \mathbf{I} - \frac{\theta}{2} [\mathbf{u}]_{\times} + \left(1 - \frac{\sin c(\theta)}{\sin c^2(\frac{\theta}{2})} \right) [\mathbf{u}]_{\times}^2 \quad (1.35)$$

La matrice d'interaction liée à la primitive

$$\mathbf{s} = \mathbf{r} = \begin{bmatrix} \mathbf{t} \\ \theta \mathbf{u} \end{bmatrix} \quad (1.36)$$

est donc donnée par :

$$\mathbf{L}_r = \begin{bmatrix} -\mathbf{I} & [\mathbf{t}]_{\times} \\ 0 & \mathbf{L}_{\omega} \end{bmatrix} \quad (1.37)$$

On peut calculer l'erreur entre les configurations courante et désirée de la caméra à partir des matrices de transformations correspondant aux deux configurations. Le calcul de la pose à partir de mesures par vision ne représente qu'une estimation qui est forcément entachée d'erreurs car la connaissance du modèle de l'objet et des paramètres de la caméra ne sont pas parfaite.

Pour assurer une décroissance exponentielle de l'erreur, le vecteur de commande est donné par :

$$\mathbf{T}_{RC/R} = -\lambda \widehat{\mathbf{L}}_r^+ \delta \hat{\mathbf{r}} \quad (1.38)$$

Où $\delta\hat{r}$ représente une estimation du vecteur d'erreur entre les deux configurations courante et désirée. Nous savons que le problème qui se pose en pratique est la reconstruction euclidienne qui s'avère indispensable à l'implémentation d'une loi de commande. Pour effectuer une reconstruction euclidienne dans des applications utilisant les robots manipulateurs, il est souhaitable de connaître le modèle de la cible par rapport à laquelle le robot doit se positionner. Par ailleurs, comme il n'existe pas de linéarité entre la transformation de l'espace projectif vers l'espace cartésien, une trajectoire rectiligne dans l'espace cartésien ne correspond pas forcément à une trajectoire rectiligne dans l'image. Il en résulte que les primitives observées peuvent sortir du champ de vue de la caméra ce qui nécessite de prendre des mesures particulières [BTG02]. Pour l'amélioration de la précision et de la robustesse de la mesure, un filtre de Kalman étendu [LWB96] ou des méthodes d'estimation s'inspirant de commande robuste telle que H1 [TO00] peuvent être implémentés. Cependant, ces solutions ne permettent pas d'éliminer les erreurs statiques ou dynamiques. En effet, comme la fonction de tâche n'est pas définie dans l'espace de mesure, une erreur nulle de la fonction de tâche ne signifie pas forcément que le robot a atteint la configuration désirée.

1.4.2.3 Asservissement visuel 2D

En asservissement visuel 2D (ou IBVS pour Image-Based Visual Servoing) la convergence se fait dans l'image. La grandeur à asservir est exprimée sous la forme de primitives visuelles dans l'image. Les informations visuelles directement extraites de l'image sont des primitives géométriques élémentaires qui proviennent de la projection de l'objet d'intérêt dans le plan image tels les coordonnées d'un point, les paramètres d'une droite, d'un segment, d'une ellipse. Afin de considérer des objets complexes, les primitives peuvent également représenter les paramètres d'un contour ou les moments 2D d'une ou les moments d'un nuage de points. Dans tous les cas, il est possible de définir l'entrée de commande s comme étant une combinaison de différents types de primitives 2D extraites de l'image. Conséquence directe, les motifs observés restent visibles au cours de l'asservissement. Pour ce type d'asservissement, on parle également d'asservissement dans le plan image ou référencé image. Contrairement à un asservissement visuel 3D, le contrôle du robot n'est pas effectué dans l'espace cartésien, mais directement dans l'image.

L'asservissement visuel 2D [BE92] consiste à contrôler les mouvements du robot de façon à déplacer les primitives observées dans l'image, de la position courante vers la position désirée. La commande est calculée à partir d'une erreur dans l'image (Voir figure 1.4). Peu de connaissance a priori du modèle de l'objet est nécessaire. Cependant, plusieurs types de primitives visuelles peuvent être utilisées. Une méthode générique a été présentée pour le calcul de la matrice

d'interaction associée à des primitives visuelles qui sont bien définies à partir des primitives géométriques (points, droites, ellipses, etc.). Dans ce cas, le signal choisi représente la configuration des primitives dans l'image notée \mathbf{s} .

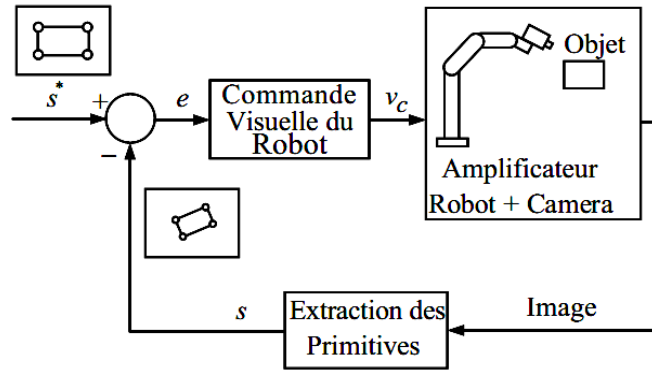


FIGURE 1.4 – Asservissement Visuel 2D.

La fonction de tâche est ainsi donnée par :

$$\mathbf{e} = \mathbf{C}(\mathbf{s} - \mathbf{s}^*) + \mathbf{L}_{2D}^+ \dot{\mathbf{s}}^* \quad (1.39)$$

On peut donc exprimer la matrice d'interaction correspondant à cette primitive comme suit :

$$\frac{d\mathbf{s}}{dt} = \frac{\partial \mathbf{s}}{\partial \mathbf{P}} \frac{d\mathbf{P}}{dt} \quad (1.40)$$

$\frac{\partial \mathbf{s}}{\partial \mathbf{P}}$ représente la jacobienne de la position du point dans l'image par rapport à celle du point dans l'espace et $\frac{d\mathbf{P}}{dt}$ représente la matrice d'interaction liée au point 3D. Dans la pratique, l'asservissement visuel 2D avec une caméra classique peut aussi bien s'effectuer à partir des coordonnées des points exprimés dans le plan normalisé qu'à partir des coordonnées des points exprimés en pixels. Dans ce cas, les coordonnées des points sont calculées à partir des coordonnées des points dans l'image par :

$$\mathbf{s} = \begin{bmatrix} \frac{1}{k_u} & 0 & -\frac{u_0}{k_u} \\ 0 & \frac{1}{k_v} & -\frac{v_0}{k_v} \end{bmatrix} \mathbf{P} \quad (1.41)$$

A partir de l'équation $\mathbf{s} = \begin{bmatrix} \frac{x}{z} \\ \frac{y}{z} \\ \frac{1}{z} \end{bmatrix}$, la dérivée temporelle de \mathbf{s} est donnée par :

$$\begin{aligned} \dot{\mathbf{s}} &= \begin{bmatrix} \frac{\dot{x}z - x\dot{z}}{z^2} \\ \frac{\dot{y}z - y\dot{z}}{z^2} \\ \frac{-\dot{z}}{z^2} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{z} & 0 & -\frac{x}{z^2} \\ 0 & \frac{1}{z} & -\frac{y}{z^2} \\ 0 & 0 & -\frac{1}{z} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{z} & 0 & -\frac{x}{z^2} \\ 0 & \frac{1}{z} & -\frac{y}{z^2} \\ 0 & 0 & -\frac{1}{z} \end{bmatrix} \dot{\mathbf{P}} \end{aligned} \quad (1.42)$$

Puis, en substituant (1.31) dans (1.42) on aboutit à :

$$\dot{\mathbf{s}} = \begin{bmatrix} \frac{1}{z} & 0 & -\frac{x}{z^2} \\ 0 & \frac{1}{z} & -\frac{y}{z^2} \\ 0 & 0 & -\frac{1}{z} \end{bmatrix} [\mathbf{I}, \quad {}^c\mathbf{OP}]_{\times} \dot{\mathbf{P}} \quad (1.43)$$

Enfin, la matrice d'interaction du point 2D est donnée par :

$$\mathbf{L}_{2D} = \begin{bmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{bmatrix} \quad (1.44)$$

z est la coordonnée suivant l'axe Z_c du repère caméra du point considéré. La matrice d'interaction est dépendante de la profondeur du point considéré. Ceci implique une connaissance de la distance entre la caméra et le point, ou le besoin de recourir à une estimation de cette distance.

À partir de l'équation (1.29) on obtient l'expression de la loi de commande qui permet une décroissance exponentielle de l'erreur :

$$\mathbf{T}_{RC/R} = -\lambda \widehat{\mathbf{L}}_{2D}^+ (\mathbf{s} - \mathbf{s}^*) + \frac{\widehat{\partial \mathbf{e}}}{\partial t} \quad (1.45)$$

Dans cette commande, l'erreur est calculée dans le même espace que la mesure. Comparativement à une commande 3D, les erreurs sur les paramètres intrinsèques ont une faible influence sur la convergence du système. Ce type de commande est robuste à une calibration des paramètres intrinsèques de la caméra. En effet, on peut noter à partir de l'équation (1.45) que si $\widehat{\mathbf{L}}_{2D}^+$ est de rang plein et pour une configuration non dégénérée du vecteur $\mathbf{s} - \mathbf{s}^*$ (le vecteur d'erreur n'étant pas contenu dans le noyau de la matrice $\widehat{\mathbf{L}}_{2D}^+$) alors $\mathbf{T}_{RC/R}$ ne peut être nulle que si l'erreur $\mathbf{s} - \mathbf{s}^*$ est nulle. La condition de convergence étant définie par l'équation (1.30). Sachant qu'il n'existe pas de linéarité entre le signal de régulation et le vecteur de commande, l'implantation d'une telle commande peut s'avérer difficile. Pour pallier à ce problème, une estimation de la matrice d'interaction à la

configuration désirée à partir de laquelle le vecteur \mathbf{s}^* a été défini est nécessaire. Cependant, comme la linéarisation de la fonction de tâche n'est que locale, une configuration trop éloignée de la configuration désirée ne permet pas de linéariser de façon exacte le système et donc d'obtenir une décroissance exponentielle de l'erreur.

1.4.2.4 Asservissement visuel hybride

L'asservissement visuel 2D1/2 (Confère figure 1.5) ou hybride tire partie des avantages des approches 3D et 2D sans en prendre leurs inconvénients. Un vecteur \mathbf{s} de dimension 6 est établi à la fois à partir de grandeurs exprimées dans le plan image et dans le repère de la caméra. La commande proposée par Malis et al. [MCB99] permet d'obtenir un découplage partiel entre les mouvements de translation et de rotation de la caméra. Elle génère une trajectoire correcte dans l'espace tout en garantissant le contrôle de la projection du centre de l'objet dans l'image afin de maintenir sa visibilité. Cette approche a donc pour objectif d'améliorer les propriétés de linéarisation et de découplage par rapport à un asservissement visuel 2D et ne requiert pas de calcul de pose. De plus, la condition de stabilité asymptotique globale du système a été démontrée [MC02a].

Dans une commande 3D, l'une des contraintes principales est la connaissance du modèle de la cible observée. Il est possible de calculer l'orientation relative de la caméra ainsi que le rapport de profondeur à partir de points coplanaires en effectuant un calcul d'homographie [HZ00] même sans connaissance a priori de la cible. Cette mesure permet donc de contrôler quatre degrés de liberté : l'orientation et la profondeur. Les deux autres degrés de liberté sont asservis sur la position d'un point dans l'image. Comme l'orientation est engendrée par le calcul d'homographie, elle est indépendante de la translation. En asservissement visuel, la robustesse et le découplage de la loi de commande sont les objectifs visés. En plus de ces deux critères, dans l'asservissement visuel 2D1/2, la loi de commande a l'avantage d'être stable. En effet, contrairement à la commande 2D, la stabilité globale de cette loi de commande est démontrée dans tout l'espace de travail pour une caméra parfaitement calibrée [MC02a]. Cependant, cette loi de commande est moins robuste face aux erreurs d'étalonnage que la commande 2D car elle comporte une reconstruction, même partielle (calcul d'homographie). De plus, la commande est plus sensible aux bruits de mesure dans l'image puisque la loi de commande utilise un seul point de référence pour commander la translation. Par ailleurs, tout comme l'asservissement visuel 3D, cette commande ne garantit pas la visibilité de tous les points dans l'image car les rotations sont estimées et contrôlées dans l'espace cartésien.

Cette technique est basée sur l'estimation de l'homographie, notée \mathbf{H} , qui relie l'image d'au moins trois points entre différents plans projectifs [Fau93]. L'homo-

graphie est une application projective bijective, correspondant à une transformation linéaire entre deux plans projectifs. Plus précisément, elle permet d'établir (à un facteur d'échelle α près) une bijection entre un objet de l'espace 3D et une image 2D, ou bien entre deux images 2D d'un même objet. Dans ce dernier cas, elle permet de lier les projections d'un ensemble de points 3D \mathbf{P}_i , appartenant à un même plan, sur deux plans image Π_1 et Π_2 par la relation :

$$\mathbf{p}_{i/\Pi_2} = \alpha \mathbf{H} \mathbf{p}_{i/\Pi_1}, \quad \forall \alpha \in \mathbf{R}^* \quad (1.46)$$

où \mathbf{p}_{i/Π_2} et \mathbf{p}_{i/Π_1} représentent respectivement la projection des points \mathbf{P}_i sur les plans Π_1 et Π_2 . Dans le cas d'un objet plan, un minimum de quatre points appariés sur les images courante et désirée permet l'estimation de l'homographie par la résolution d'un système linéaire. Dans le cas d'un objet quelconque, un appariement de huit points est nécessaire.

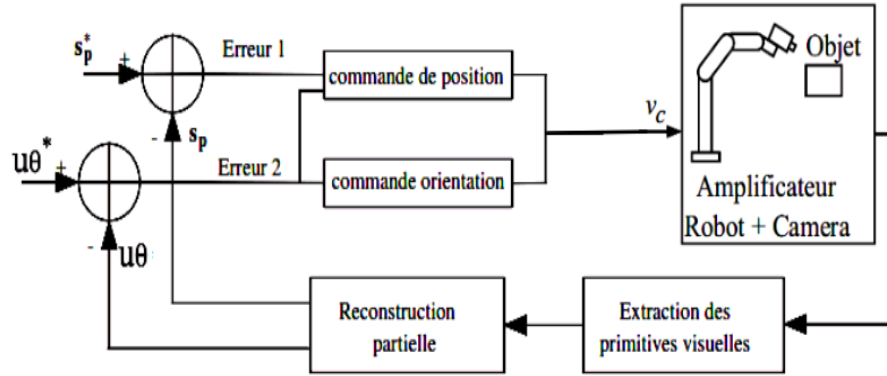


FIGURE 1.5 – Asservissement Visuel hybride.

Ainsi, l'asservissement visuel 2D1/2 repose sur la détermination de l'homographie \mathbf{H} entre l'image courante et l'image désirée. En effet, à partir de cette homographie il est possible de calculer le déplacement en rotation \mathbf{R} que la caméra doit effectuer pour atteindre la situation spécifiée, ainsi que la direction de son déplacement en translation. Cette matrice \mathbf{H} fournit également le rapport z/z^* entre les distances courante et désirée de la caméra à l'objet. Comme la translation à réaliser n'est connue qu'à un facteur d'échelle α près, un asservissement visuel purement 3D est impossible. Cependant, il est possible d'aboutir à une solution grâce à la combinaison des informations 2D (fournies par l'image) et 3D disponibles. Le vecteur des informations visuelles \mathbf{s} est donc défini sur la base de donnée 2D et 3D. La boucle d'asservissement ainsi synthétisée permet de séparer la rotation et la translation de la caméra, et d'obtenir un fort découplage de la loi de commande. L'avantage majeur de cette approche est que la connaissance du modèle géométrique 3D de l'objet n'est plus nécessaire. La seule information 3D

utilisée dans la synthèse de la commande est la profondeur désirée z^* d'un point de l'objet.

1.4.2.5 Discussion

Différentes solutions ont été proposées pour estimer la pose dans le cas d'un capteur monoculaire. Ces solutions utilisent des primitives extraites de l'image et du modèle géométrique de l'objet, en connaissant les paramètres intrinsèques de la caméra. La méthode de Tsai [Tsa86] permet de reconstruire la pose d'un objet en utilisant n points coplanaires ; la méthode de Dementhon [DeM92] utilise aussi n points, qui peuvent ne pas être coplanaires, mais sa résolution est itérative ce qui peut être problématique dans une boucle de commande à période d'échantillonnage fixe. D'autres méthodes sont basées sur l'utilisation d'une résolution par moindres carrés [Low91]. Récemment, une approche basée sur l'asservissement visuel virtuel a été proposée par Marchand et al [MC02b], en considérant le problème d'estimation de pose comme un problème dual de l'asservissement visuel dans l'image.

L'inconvénient majeur de l'asservissement visuel 3D est l'estimation de la pose. Si l'on utilise des algorithmes itératifs [Low91, DeM92] pour estimer la pose, le temps de calcul peut être important et ne pas respecter les contraintes temps réel. De plus, ces méthodes nécessitent d'avoir un modèle de l'objet. Quelle que soit la méthode utilisée, itérative ou basée sur la géométrie épi polaire (géométrie d'un système à deux prises de vue), l'étape d'estimation entraîne une forte sensibilité de l'architecture aux erreurs de calibration des paramètres de la caméra et aux bruits de mesure. Par contre son avantage repose sur l'obtention de la pose cartésienne de la caméra ou de l'effecteur du robot qui permet d'obtenir des trajectoires cartésiennes réalisables.

L'avantage majeur de l'asservissement visuel dans l'image est sa robustesse par rapport aux erreurs de calibration du capteur visuel [Esp93]. En contrepartie, puisque l'asservissement est réalisé dans l'espace du capteur, les trajectoires cartésiennes correspondantes du robot ne sont pas contraintes, et certains mouvements désirés dans l'image peuvent conduire à des trajectoires inadéquates dans l'espace cartésien ou articulaire. Durant un asservissement visuel, les primitives visuelles doivent rester visibles pendant toute la durée du mouvement. Dans la structure présentée précédemment, la profondeur des primitives observées z n'est pas accessible à partir de l'image seule. On peut alors utiliser une estimation de z en position désirée et la matrice d'interaction est ainsi calculée à convergence. La visibilité des primitives visuelles n'est alors pas assurée. Cette situation se présente généralement lorsque le mouvement entre les positions initiale et désirée est important. Par exemple, pour un mouvement de rotation autour de l'axe X_m ou Y_m du repère de l'objet, les déplacements induits par le contrôleur peuvent

conduire à la perte de visibilité de certaines primitives visuelles au cours du mouvement. La trajectoire cartésienne obtenue lors d'un asservissement visuel dans l'image peut être non optimale [Mor99]. En effet, lorsque la matrice d'interaction est estimée à chaque itération et que les paramètres intrinsèques sont correctement estimés, le mouvement des primitives visuelles dans l'image est rectiligne. Or ce mouvement dans l'image ne se traduit pas forcément par un mouvement rectiligne de l'effecteur du robot dans l'espace cartésien. Différentes approches ont été proposées dans la littérature pour pallier ou éviter les problèmes de l'asservissement visuel dans l'image. Le lecteur intéressé pourra se référer à Nelson et al [Nel95].

Dans l'asservissement visuel 2D1/2 proposé par Malis et al [MCB99], une reconstruction partielle de la pose de la caméra, basée sur la décomposition de la matrice d'homographie calculée entre deux images consécutives, permet d'exprimer le contrôle de la position dans l'espace image, et le contrôle de l'orientation dans l'espace cartésien. Une propriété intéressante de cette structure de commande est le découplage entre la commande en rotation et la commande en translation. L'utilisation de la décomposition de la matrice d'homographie a aussi été explorée dans [Deg98]. Malheureusement, la décomposition de cette matrice rend le système sensible aux bruits de mesure. De plus, il est nécessaire d'avoir de l'information supplémentaire pour sélectionner la solution parmi les solutions analytiques possibles, mais ce problème peut facilement être contourné en utilisant une troisième image.

1.4.2.6 Conclusion

Dans le cadre de l'asservissement visuel, le choix des primitives [TFC09, NAH02] et des paramètres les décrivant [IO05, KI07] est primordial afin de réaliser la synthèse de lois de commande permettant d'obtenir les meilleures propriétés de linéarisation et de découplage. Dans les applications de commande de robot, la fonction de tâche définit un critère à optimiser qui s'apparente dans le contexte de l'optimisation non linéaire à la fonction de coût. Dans l'asservissement visuel 2D classique, cette fonction représente l'erreur entre la position des primitives visuelles observées et désirées dans l'image, transformées par la matrice de combinaison en un vecteur à l éléments correspondant aux l degrés de liberté à contrôler. Ainsi, dans les applications où le mouvement est défini dans l'espace, la dimension du vecteur calculé par la fonction de tâche est de 6×1 . En réalité, tous les modèles de projection utilisés dans l'élaboration de schémas de commande par vision utilisent des hypothèses qui permettent de simplifier les lois de commande obtenues.

Dans le chapitre suivant, nous exposons l'application des moments 2D extraits de l'image comme primitives visuelles que nous intégrons dans une loi de

commande en l'asservissement visuel dans l'image. Nous détaillons la modélisation utilisée, et nous présentons les résultats obtenus en termes d'informations visuelles pour commander les six degrés de liberté de la caméra.

Chapitre 2

Modélisation

2.1 Introduction

Ce chapitre est essentiellement focalisé sur le calcul des primitives visuelles \mathbf{s} . Afin de développer une méthode reposant sur le lien entre la variation des informations visuelles $\dot{\mathbf{s}}$ et le mouvement de la caméra $\mathbf{T}_{RC/R}$ défini par la relation (1.29) que nous avons présentée au chapitre I, la stratégie envisagée consiste à exploiter l'historique des mouvements de la caméra et des mesures des indices visuels \mathbf{s} . Dans la suite de ce chapitre, le cas général d'un système à six degrés de liberté est utilisé et $\mathbf{T}_{RC/R}$ est alors défini par :

$$\mathbf{T}_{RC/R} = (\mathbf{v}, \omega) \quad (2.1)$$

Où $\mathbf{v} = (v_x, v_y, v_z)$ et $\omega = (\omega_x, \omega_y, \omega_z)$ sont respectivement les vitesses de translations et de rotation du mouvement de la caméra.

Un état de l'art sur les informations visuelles les plus utilisées est fait dans ce chapitre. Et pour chaque primitive, nous faisons bien de rappeler les primitives visuelles existantes. Par la suite un nouvel ensemble d'informations visuelles est donné en exploitant les moments géométriques 2D non centrés.

2.2 Mouvement de la caméra et variation des informations visuelles

Considérons un objet \mathbf{O} rigide et fixe de la scène. Ce solide se projette sur le plan image de la caméra en une région \mathbf{R} (cf. figure 2.1) à laquelle est associée un ensemble de primitives visuelles \mathbf{s} , et un vecteur \mathbf{z} décrivant leur profondeur.

Comme nous l'avons préalablement introduit, la variation des informations visuelles $\dot{\mathbf{s}}$ est liée à la cinématique de la caméra $\mathbf{T}_{RC/R} = (\mathbf{v}, \omega)$ par le biais de

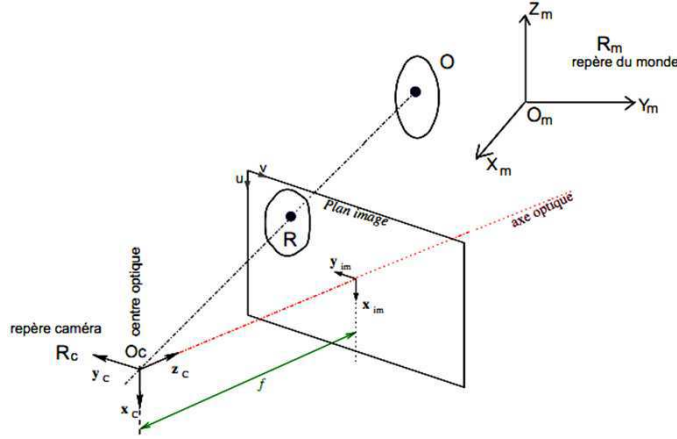


FIGURE 2.1 – Projection d'un objet quelconque dans l'image.

la matrice d'interaction \mathbf{L}_s :

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{T}_{RC/R} \quad (2.2)$$

2.2.1 Primitive Point

Le point est une primitive couramment employée en asservissement visuel, car il s'agit d'une primitive très simple que l'on sait aisément extraire de l'image à l'aide d'algorithmes efficaces. De manière très classique, le modèle mathématique d'une caméra est défini alors par une projection perspective telle que tout point \mathbf{P} ayant pour coordonnées $\mathbf{X} = (X, Y, Z)$ se projette sur le plan image en un point \mathbf{p} de coordonnées $\mathbf{X} = (x, y)$ avec :

$$\begin{cases} x = X/Z \\ y = Y/Z \end{cases} \quad (2.3)$$

Le vecteur des informations sensorielles \mathbf{s} est alors défini par $\mathbf{s} = (x, y)^T$. Comme nous l'avons vu au chapitre I, en vision perspective, la variation $\dot{\mathbf{s}}$ d'un point \mathbf{p} s'écrit classiquement [Cha90] :

$$\dot{\mathbf{s}} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 1/Z & 0 & -X/Z^2 \\ 0 & 1/Z & -Y/Z^2 \end{pmatrix} \dot{\mathbf{X}} \quad (2.4)$$

Quelle que soit la configuration choisie (caméra embarquée ou déportée, point \mathbf{P} fixe ou mobile), la vitesse $\dot{\mathbf{X}}$ de \mathbf{P} en fonction du torseur cinématique $\mathbf{T}_{RC/R}$

entre la camera et son environnement est donnée par l'équation fondamentale de la cinématique :

$$\dot{\mathbf{X}} = -v - \omega \times \mathbf{X} = -\mathbf{v} + [\mathbf{X}]_{\times} \omega = \begin{pmatrix} -\mathbf{I}_3 & [\mathbf{X}]_{\times} \end{pmatrix} \mathbf{T}_{RC/R} \quad (2.5)$$

Cette équation peut alors se simplifier sous la forme :

$$\dot{\mathbf{X}} = \mathbf{L}_s \mathbf{T}_{RC/R} \quad (2.6)$$

où \mathbf{L}_s est la matrice d'interaction dépendant de la profondeur Z inconnue dans ce cas, donnée par la relation ci-après :

$$\mathbf{L}_s = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \quad (2.7)$$

2.2.2 Les autres primitives géométriques usuelles

Bien qu'il soit possible de construire différentes primitives géométriques sur la base du point, il est parfois intéressant de considérer d'autres paramétrisations du vecteur des indices visuels \mathbf{s} pour définir la tâche référencée vision. Nous nous intéressons ici aux primitives géométriques usuelles telles que les droites, les ellipses, les sphères, les cylindres, etc.

Considérons un objet \mathbf{O} fixe de la scène qui se projette dans l'image en une région \mathbf{R} ayant un contour fermé C (cf. figure 2.1). Nous admettons que cet objet est formé d'une surface continue, c'est-à-dire ne présentant pas de discontinuité sur la profondeur. Dans ce cas, la profondeur z de chaque point de l'objet \mathbf{O} peut s'exprimer par une fonction continue des points images $\mathbf{P}_j(X_j, Y_j) \in \mathbf{R}$ de la manière suivante [Cha90, Cha04], [BE92] :

$$\frac{1}{z_j} = \sum_{p \geq 0, q \geq 0} \mathbf{A}_{pq} X_j^p Y_j^q, \quad \forall X_j, Y_j \in \mathbf{R} \quad (2.8)$$

où les coefficients \mathbf{A}_{pq} dépendent de la nature de l'objet \mathbf{O} considéré. Par exemple, dans le cas d'un objet plan dont l'équation s'exprime dans le repère caméra \mathbf{R}_C comme suit :

$$z_j = ax_j + by_j + c, \forall p_j : (x_j, y_j, z_j) \in \mathbf{O} \quad (2.9)$$

l'équation (2.8) se réduit à [Cha04] :

$$\frac{1}{z_j} = AX_j + BY_j + C \quad (2.10)$$

avec $C = \mathbf{A}_{00} = \frac{1}{c}$, $A = \mathbf{A}_{10} = -\frac{a}{c}$, $B = \mathbf{A}_{01} = -\frac{b}{c}$
et $\mathbf{A}_{pq} = 0 \quad \forall (p, q) = (1, 1) \quad \text{et } p, q > 1$

2.2.2.1 Les moments 2D de l'image

Les moments de l'image sont des informations qui ont l'avantage d'être génériques quant à la caractérisation de formes 2D très variées. Les moments de faible ordre ont un sens géométrique intuitif puisqu'ils représentent directement l'aire, le barycentre et l'orientation principale dans l'image de l'objet d'intérêt. Etant obtenus par intégration et plus particulièrement par une somme discrétisée sur la section observée dans l'image, ils ont l'avantage d'être peu sensibles aux bruits de mesure. Les moments de l'image ont été largement utilisés dans le domaine de la vision par ordinateur notamment pour des applications de reconnaissance de formes, où des fonctions de moments invariantes aux mouvements de translation et rotation dans l'image ont été élaborées. Ils ont également été introduits en asservissement visuel, où 6 primitives visuelles construites à partir d'une combinaison des moments 2D ont été proposées dans la commande du système pour réaliser la tâche de positionnement d'une caméra [TC05a, Cha03].

Les moments de l'image constituent donc un autre ensemble de primitives plus récemment considérées en asservissement visuel et qui offrent un certain nombre de propriétés intéressantes. Tout d'abord, ils fournissent une représentation générique et significative des objets de forme simple ou complexe en proposant une approche statistique. Ensuite, ils présentent l'avantage de traiter la forme de l'objet dans son ensemble, de sorte qu'il n'est plus nécessaire de chercher à mettre en correspondance les points d'une image à l'autre, mais uniquement des régions. Enfin, les moments permettent d'obtenir de très bonnes propriétés de découplage du mouvement de la caméra (ie. séparation des mouvements de translation et de rotation) [Tah04, TC05a]. Par conséquent, afin de rendre possible l'utilisation des moments 2D dans un asservissement visuel, il s'agit, de modéliser la matrice d'interaction permettant d'exprimer leur variation en fonction du torseur cinématique de la caméra. Nous nous proposons de rappeler brièvement les définitions relatives aux moments 2D ainsi qu'aux moments centrés de l'image. Comme précédemment, nous considérons un objet \mathbf{O} fixe de la scène qui se projette en une région \mathbf{R} dans l'image ayant un contour fermé C (cf. figure 2.1). Afin de simplifier les expressions, nous supposons, dans cette section, que l'objet \mathbf{O} observé est plan. La profondeur z des points appartenant à \mathbf{O} est alors reliée aux coordonnées $(x_j, y_j)^T$ des points de l'image par la relation (2.3).

2.2.2.2 Définitions et propriétés des moments classiques 2D

Le moment 2D de l'image m_{ij} d'ordre $i + j$ de l'objet \mathbf{O} est classiquement défini pour tous les points $\mathbf{p}(x, y)$ de la région \mathbf{R} de l'image par la relation suivante :

$$m_{pq} = \iint_R x^p y^q f(x, y) dx dy \quad (2.11)$$

Où $f(x, y)$ est le niveau de gris d'un point de coordonnées (x, y) dans le cas d'une image.

- Moment d'ordre zéro

Le moment d'ordre zéro m_{00} d'une distribution $f(x, y)$ est donnée par :

$$m_{00} = \iint_R f(x, y) dx dy \quad (2.12)$$

Il représente la masse totale de la distribution $f(x, y)$. Dans le cas d'un objet binaire, m_{00} n'est autre que la surface de l'objet.

- Moments d'ordre 1

Les moments d'ordre 1, c'est-à-dire m_{10} et m_{01} , permettent de déterminer la position du centre de gravité de l'objet. Plus précisément, si on note x_g et y_g les coordonnées du centre de gravité de l'objet, on a :

$$\begin{cases} x_g = \frac{m_{10}}{m_{00}} \\ y_g = \frac{m_{01}}{m_{00}} \end{cases} \quad (2.13)$$

- Moments centrés

Les moments centrés μ_{pq} sont calculés à partir du centre de gravité (x_g, y_g) . Ils sont définis par :

$$\mu_{pq} = \iint_R (x - x_g)^p (y - y_g)^q f(x, y) dx dy \quad (2.14)$$

Les moments centrés peuvent être calculés directement à partir des moments m_{pq} d'ordre inférieur à $p+q+1$ par :

$$\mu_{pq} = \sum_{k=0}^p \sum_{l=0}^q \binom{p}{k} \binom{q}{l} (-x_g)^{p-k} (-y_g)^{q-l} m_{kl} \quad (2.15)$$

Réciproquement, il est possible de déterminer les moments à partir des moments centrés. On a en effet :

$$m_{pq} = \sum_{k=0}^p \sum_{l=0}^q \binom{p}{k} \binom{q}{l} (x_g)^{p-k} (y_g)^{q-l} \mu_{kl} \quad (2.16)$$

- Moments centrés d'ordre 2

Les moments centrés d'ordre 2 (μ_{20} , μ_{11} , et μ_{02}) sont appelés les moments d'inertie. Avec les coordonnées du centre de gravité, ils permettent de définir une ellipse dont les moments d'ordre inférieur ou égal à 2 ont la même valeur que ceux de l'objet (voir Figure 2.2). L'équation de cette ellipse est donnée par :

$$\frac{(x - x_g + t(y - y_g))^2}{a_1^2(1 + t^2)} + \frac{(y - y_g + t(x - x_g))^2}{a_2^2(1 + t^2)} - 1 = 0 \quad (2.17)$$

Où a_1 et a_2 représentent respectivement les deux axes principaux majeur et mineur et où t est liée à l'orientation de l'objet. Plus précisément, on a

$$\theta = \frac{1}{2} \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (2.18)$$

A partir des paramètres de l'ellipse, on peut facilement exprimer les moments centrés d'ordre 2. On a en effet :

$$\begin{cases} \mu_{20} = m_{00} ((a_1^2 + a_2^2 t^2) / 4 (1 + t^2)) \\ \mu_{11} = m_{00} (t (a_1^2 - a_2^2) / 4 (1 + t^2)) \\ \mu_{02} = m_{00} ((a_2^2 + a_1^2 t^2) / 4 (1 + t^2)) \end{cases} \quad (2.19)$$

Où

$$m_{00} = \pi a_1 a_2 \quad (2.20)$$

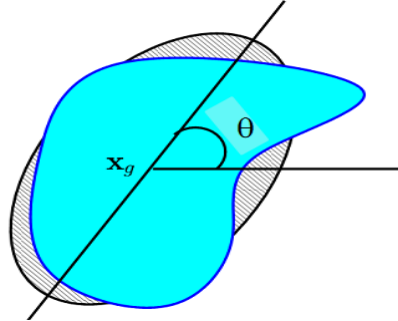


FIGURE 2.2 – Orientation d'un objet et définition des paramètres usuels d'une ellipse.

Réciproquement, les paramètres de l'ellipse peuvent être déterminés à partir des équations suivantes :

$$\begin{cases} a_1^2 = \frac{2}{\mu_{00}} \left(\mu_{02} + \mu_{20} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \right) \\ a_2^2 = \frac{2}{\mu_{00}} \left(\mu_{02} + \mu_{20} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \right) \\ t = \left(\mu_{02} + \mu_{20} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \right) / 2\mu_{11} \end{cases} \quad (2.21)$$

- Moments centrés d'ordre 3

Les moments centrés d'ordre 3 évaluent le degré d'asymétrie d'un objet par rapport à ses axes principaux. En reconnaissance des formes, on utilise souvent les coefficients d'asymétrie définis de la manière suivante : $S_x = \frac{\mu_{30}}{\mu_{20}^{3/2}}$ et $S_y = \frac{\mu_{03}}{\mu_{02}^{3/2}}$. Les moments d'ordre 3 permettent également de résoudre le problème du choix de la valeur de θ . En effet, on peut noter à partir de la figure 2.2 qu'il y a deux solutions pour l'orientation de l'objet : θ et $\theta + \pi$. Les moments d'ordre 3, lorsqu'ils ne sont pas nuls, peuvent être employés pour résoudre cette ambiguïté. En effet, une rotation d'un objet de 180 degrés change le signe des moments d'ordre 3. Pour distinguer les deux solutions de θ , on peut fixer le sens des axes principaux dans une image de référence.

2.2.2.3 Matrices d'interaction associées aux moments classiques 2D :

Transformation d'intégrales de régions en intégrales de contours et vis versa

Dans ce qui suit on suppose que l'image est binaire, ou que le niveau de gris ne change pas si la caméra bouge. Soit $\Omega_{\text{int}} \subset \mathbb{R}^2$ un contour fermé définissant une région et $f(x, y) : \Omega_{\text{int}} \rightarrow \mathbb{R}$ une image dans Ω_{int} , la région contenant les images. L'objectif est de calculer la variation temporelle de m_{pq} défini par :

$$m_{pq} = \iint_{\Omega_{\text{int}}} k(x, y, \Omega_{\text{int}}) dx dy \quad (2.22)$$

où le descripteur de la région est défini par l'expression :

$$k(x, y, \Omega_{\text{int}}) = x^p y^q f(x, y) \quad (2.23)$$

Pour calculer la variation temporelle de m_{pq} , nous pouvons utiliser le théorème de Green-Riemann [Cha03] ou bien une approche basée sur la dérivée Eulérienne [JS03]. Soit $(\mathbf{F}(t, \cdot))_{t \geq 0}$ une famille de transformations bijectives. Pour un point $\mathbf{X} = [x, y]^T$ on note :

$$\mathbf{X}(t) = \mathbf{F}(t, \mathbf{X}) \quad (2.24)$$

où $\mathbf{X}(0) = \mathbf{X}$ et $\Omega_{in}(t) = \mathbf{F}(t, \Omega_{in})$. Soit $\dot{\mathbf{X}}$ la vitesse d'un point du contour $\mathbf{X} = [x, y]^T$:

$$\dot{\mathbf{X}}(t, \mathbf{X}(t)) = \frac{\partial \mathbf{F}}{\partial t}(t, \mathbf{X}) \quad (2.25)$$

Comme nous nous intéressons à de petites déformations pour le calcul de la dérivée, en utilisant le théorème d'expansion de Taylor, on a :

$$\mathbf{F}(t, \mathbf{X}) = \mathbf{F}(0, \mathbf{X}) + t \frac{\partial \mathbf{F}}{\partial t}(0, \mathbf{X}) \quad (2.26)$$

La dérivée Eulérienne dans la direction du champ de vecteurs $\dot{\mathbf{X}}$ (vitesse d'un point $\mathbf{X}(x, y)$) nous donne

$$\dot{m}_{pq} = \lim_{t \rightarrow 0} \frac{(m_{pq}(\Omega_{int}(t)) - m_{pq}(\Omega_{int}))}{t} \quad (2.27)$$

La dérivée matérielle de $k(\mathbf{X}, \Omega_{int})$:

$$\dot{k} = \lim_{t \rightarrow 0} \frac{(k(\mathbf{X}(t), \Omega_{int}(t)) - k(\mathbf{X}, \Omega_{int}))}{t} \quad (2.28)$$

La dérivée de domaine de $k(\mathbf{X}, \Omega_{int})$ est définie par :

$$k' = \lim_{t \rightarrow 0} \frac{(k(\mathbf{X}, \Omega_{int}(t)) - k(\mathbf{X}, \Omega_{int}))}{t} \quad (2.29)$$

De toute évidence, en transformant (3, 27) selon la première formule de Taylor, nous avons :

$$\dot{k}(\mathbf{X}, \Omega_{int}) = k'(\mathbf{X}, \Omega_{int}) + \nabla k(\mathbf{X}, \Omega_{int}) \cdot \dot{\mathbf{X}} \quad (2.30)$$

La transformation de m_{pq} donne :

$$\frac{m_{pq}(\Omega_{int}(t)) - m_{pq}(\Omega_{int})}{t} = \frac{1}{t} \left[\int_{\Omega_{int}(t)} k(\mathbf{X}, \Omega_{int}(t)) d\mathbf{X} - \int_{\Omega_{int}} k(\mathbf{X}, \Omega_{int}) d\mathbf{X} \right] \quad (2.31)$$

Dans la première intégrale on fait un changement de variable de telle sorte que :

$$\mathbf{X}(t) = \mathbf{X} + t\dot{\mathbf{X}} \text{ où}$$

$$\dot{\mathbf{X}} = \left[\dot{\mathbf{X}}_1(x, y), \dot{\mathbf{X}}_2(x, y) \right]^T \quad (2.32)$$

Qui donne :

$$\int_{\Omega_{int}(t)} k(\mathbf{X}, \Omega_{int}(t)) d\mathbf{X} = \int_{\Omega_{int}} k(\mathbf{X} + tV, \Omega_{int} + tV) |\det \mathbf{J}(\mathbf{X})| d\mathbf{X} \quad (2.33)$$

où $\mathbf{J}(\mathbf{X})$ est la matrice jacobienne :

$$\mathbf{J}(\mathbf{X}) = \begin{bmatrix} 1 + t \frac{\partial \dot{\mathbf{X}}_1}{\partial x} & t \frac{\partial \dot{\mathbf{X}}_1}{\partial y} \\ t \frac{\partial \dot{\mathbf{X}}_2}{\partial x} & 1 + t \frac{\partial \dot{\mathbf{X}}_2}{\partial y} \end{bmatrix} \quad (2.34)$$

On a donc :

$$\det J(\mathbf{X}) = 1 + t \operatorname{div}(\dot{\mathbf{X}}) + \Theta(t) \quad (2.35)$$

Enfin on a :

$$\lim_{t \rightarrow 0} \frac{\det \mathbf{J}(\mathbf{X}) - 1}{t} = \operatorname{div}(\dot{\mathbf{X}}) \quad (2.36)$$

L' équation (2.36) devient :

$$\frac{m_{pq}(\Omega_{int}(t)) - m_{pq}(\Omega_{int})}{t} = I_1 + I_2 \quad (2.37)$$

avec :

$$\lim_{t \rightarrow 0} I_1 = \int_{\Omega_{int}} \dot{k}(\mathbf{X}, \Omega_{int}) d\mathbf{X} = \int_{\Omega_{int}} k'(\mathbf{X}, \Omega_{int}) d\mathbf{X} + \int_{\Omega_{int}} \nabla k(\mathbf{X}, \Omega_{int}) \cdot \dot{\mathbf{X}} d\mathbf{X} \quad (2.38)$$

$$\lim_{t \rightarrow 0} I_2 = \int_{\Omega_{int}} k(\mathbf{X}, \Omega_{int}) \operatorname{div}(\dot{\mathbf{X}}) d\mathbf{X} \quad (2.39)$$

On obtient finalement pour la dérivée Eulérienne :

$$\dot{m}_{pq} = \int_{\Omega_{int}} k'(\mathbf{X}, \Omega_{int}) d\mathbf{X} + \int_{\Omega_{int}} \operatorname{div}(k(\mathbf{X}, \Omega_{int}) \dot{\mathbf{X}}) d\mathbf{X} \quad (2.40)$$

Comme notre descripteur ne dépend pas de la région ie

$$k(x, y, \Omega_{int}) = k(\mathbf{X}) \quad (2.41)$$

On obtient :

$$\dot{m}_{pq} = \int_{\Omega_{int}} \operatorname{div}(k(\mathbf{X}, \Omega_{int}) \dot{\mathbf{X}}) d\mathbf{X} \quad (2.42)$$

En développant (2.42), nous obtenons finalement

$$\dot{m}_{pq} = \iint_{\Omega_{int}} \left[\frac{\partial k}{\partial x} \dot{x} + \frac{\partial k}{\partial y} \dot{y} + k(x, y) \left(\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} \right) \right] dx dy \quad (2.43)$$

La même relation a été obtenue dans [TC05a, Cha03] , mais en utilisant le théorème de Green-Riemann.

La variation \dot{m}_{pq} est alors donnée par [Cha04] :

$$\dot{m}_{pq} = \mathbf{L}_{m_{pq}} \begin{bmatrix} \mathbf{v} \\ \omega \end{bmatrix} \quad (2.44)$$

où $\mathbf{L}_{m_{pq}}$ est la matrice d'interaction calculée à partir de m_{pq} , et où le torseur cinématique s'écrit $\mathbf{T}_{RC/R} = (\mathbf{v}, \omega)$ où $\mathbf{v} = (v_x, v_y, v_z)$ et $\omega = (\omega_x, \omega_y, \omega_z)$ sont respectivement les vitesses de translations et de rotation du mouvement de la caméra.

Les expressions \dot{x} , \dot{y} , $\frac{\partial \dot{x}}{\partial x}$ et $\frac{\partial \dot{y}}{\partial y}$ dans (2.43) s'écrivent en fonction des composantes de $\mathbf{T}_{RC/R}$ en utilisant la relation (2.10) et la matrice d'interaction associée au point (2.7) :

$$\begin{cases} \dot{x} = -(Ax + By + C)v_x + x(Ax + By + C)v_z \\ \quad + xy\omega_x - (1 + x^2)\omega_y + y\omega_z \\ \dot{y} = -(Ax + By + C)v_y + y(Ax + By + C)v_z \\ \quad + (1 + y^2)\omega_x - xy\omega_y - x\omega_z \end{cases} \quad (2.45)$$

A partir de (2.45) on en déduit :

$$\begin{cases} \frac{\partial \dot{x}}{\partial x} = -Av_x + (Ax + By + C)v_z + y\omega_x - 2x\omega_y \\ \frac{\partial \dot{y}}{\partial y} = -Bv_y + (Ax + By + C)v_z + 2y\omega_x - x\omega_y \end{cases} \quad (2.46)$$

De [TC05a] , on en déduit la matrice d'interaction $\mathbf{L}_{m_{pq}}$ comme suit :

$$\mathbf{L}_{m_{pq}} = \begin{bmatrix} m_{v_x} & m_{v_y} & m_{v_z} & m_{w_x} & m_{w_y} & m_{w_z} \end{bmatrix} \quad (2.47)$$

avec :

$$\begin{cases} m_{v_x} = -p(Am_{pq} + Bm_{p-1,q+1} + Cm_{p-1,q}) - Am_{pq} \\ m_{v_y} = -q(Am_{p+1,q-1} + Bm_{pq} + Cm_{p,q-1}) - Bm_{pq} \\ m_{v_z} = (p + q + 3)(Am_{p+1,q} + Bm_{p,q+1} + Cm_{pq}) - Cm_{pq} \\ m_{w_x} = (p + q + 3)m_{p,q+1} + qm_{p,q-1} \\ m_{w_y} = -(p + q + 3)m_{p+1,q} - pm_{p-1,q} \\ m_{w_z} = pm_{p-1,q+1} - qm_{p+1,q-1} \end{cases}$$

Notons que la matrice d'interaction $\mathbf{L}_{m_{pq}}$ dépend des moments d'ordre inférieur ou égal à $p + q + 1$. Enfin, $\mathbf{L}_{m_{pq}}$ est une fonction implicite de la profondeur $\mathbf{z}(\mathbf{q}, t)$ de l'objet \mathbf{O} observé, puisqu'elle nécessite la détermination des composantes A , B et C définissant l'équation du plan.

De même, la variation $\dot{\mu}_{pq}$ des moments centrés se déduit de celle des moments classiques par la relation ci-après [Cha04] :

$$\dot{\mu}_{pq} = \mathbf{L}_{\mu_{pq}} \begin{bmatrix} \mathbf{v} \\ \omega \end{bmatrix} \quad (2.48)$$

où $\mathbf{L}_{\mu_{pq}}$ s'écrit [TC05a] :

$$\mathbf{L}_{\mu_{pq}} = \begin{bmatrix} \mu_{v_x} & \mu_{v_y} & \mu_{v_z} & \mu_{w_x} & \mu_{w_y} & \mu_{w_z} \end{bmatrix} \quad (2.49)$$

avec :

$$\begin{cases} \mu_{v_x} = -(p+1) A \mu_{pq} - p B \mu_{p-1,q+1} \\ \mu_{v_y} = -q A \mu_{p+1,q-1} - (q+1) B \mu_{pq} \\ \mu_{v_z} = -A \mu_{w_y} + B \mu_{w_x} + (p+q+2) C \mu_{pq} \\ \mu_{w_x} = (p+q+3) \mu_{p,q+1} + p x_g \mu_{p-1,q+1} \\ \quad + (p+2q+3) y_g \mu_{pq} - p^4 n_{11} \mu_{p-1,q} - q^4 n_{02} \mu_{p,q-1} \\ \mu_{w_y} = -(p+q+3) \mu_{p+1,q} - (2p+q+3) x_g \mu_{pq} \\ \quad - q y_g \mu_{p+1,q-1} + p^4 n_{20} \mu_{p-1,q} + q^4 n_{11} \mu_{p,q-1} \\ \mu_{w_z} = p \mu_{p-1,q+1} - q \mu_{p+1,q-1} \end{cases}$$

et

$$n_{pq} = \frac{\mu_{pq}}{m_{00}}$$

2.2.2.4 Les moments 2D de l'image "shifted"

[Hu62] a utilisé les moments centraux pour obtenir l'invariance aux translations. Seulement, les images ayant une symétrie le long des axes x et y et symétrique par rapport au centre de gravité de l'objet donnent des valeurs nulles pour les moments centraux d'ordres impairs. Cela est dû à l'emplacement symétrique des pixels de l'image autour du centre de gravité. Dans le présent document, nous proposons un nouveau type de moments géométriques calculés à partir d'un point autre que le centre de gravité de l'image. Ce nouveau centre est choisi de telle sorte que les moments soient invariants à de nouvelles propriétés telles que la translation, l'échelle et la rotation. Notre approche permet d'obtenir des valeurs différentes de zéro pour tous les moments d'ordre pairs et impairs et résout ainsi les problèmes causés par des images symétriques. En raison du fait que le nouveau centre est décalé par rapport au centre de gravité de l'image, tous les pixels vont contribuer au calcul des moments dans l'image.

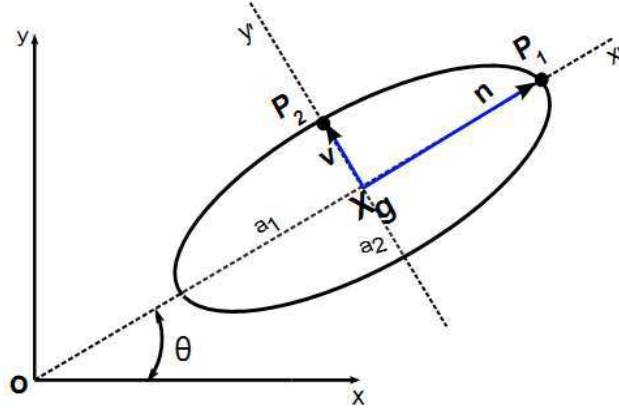


FIGURE 2.3 – Représentation de l'image.

Définitions et propriétés des moments 2D "shifted"

Définissons les nouveaux moments "shifted" comme suit :

$$\mu_{pq}^{sh} = \iint_R (x - x_g + x_{sh})^p (y - y_g + y_{sh})^q f(x, y) dx dy \quad (2.50)$$

où les paramètres "shifted" x_{sh} et y_{sh} sont définis de la manière suivante :

- En prenant le point P_1

Dans la direction de \mathbf{n} , on a (confère figure 2.3)

$$\begin{cases} x_{sh1} = x_g + a \cos \theta \\ y_{sh1} = y_g + a \sin \theta \end{cases} \quad (2.51)$$

- En prenant le point P_2

Dans la direction de \mathbf{v} , on a (confère figure 2.3)

$$\begin{cases} x_{sh2} = x_g + a \cos \left(\theta + \frac{\pi}{2} \right) \\ y_{sh2} = y_g + a \sin \left(\theta + \frac{\pi}{2} \right) \end{cases} \quad (2.52)$$

où le paramètre " a " peut s'écrire comme suit :

$$a = (\mu_{20} + \mu_{02})^{\frac{1}{4}} \quad (2.53)$$

Si on ne veut pas faire de tracking il suffit de prendre comme paramètres les axes principaux

$$\begin{cases} a_1^2 = \frac{2}{\mu_{00}} \left(\mu_{02} + \mu_{20} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \right) \\ a_2^2 = \frac{2}{\mu_{00}} \left(\mu_{02} + \mu_{20} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \right) \end{cases} \quad (2.54)$$

Pour déterminer l'invariance aux translations de μ_{pq}^{sh} , soit la fonction image définie de la manière suivante

$$g(x, y) = f(x - x_0, y - y_0) \quad (2.55)$$

$g(x, y)$ est la fonction décalée de $f(x, y)$. En remplaçant (2.55) dans (2.50), il est évident de voir que μ_{pq}^{sh} est la même pour les deux fonctions $f(x, y)$ et $g(x, y)$ et par conséquent, μ_{pq}^{sh} est invariant à la translation.

Les nouveaux moments non centrés peuvent être normalisés et devenir invariant à l'échelle de la manière suivante,

$$\eta_{pq}^{sh} = \frac{\mu_{pq}^{sh}}{m_{00}^{(p+q+2)/2}} \quad (2.56)$$

Pour montrer l'invariance à l'échelle des nouveaux moments, définissons

$$h(x, y) = f\left(\frac{x}{b}, \frac{y}{b}\right) \quad (2.57)$$

Où $h(x, y)$ représente un changement d'échelle d'une image donnée $f(x, y)$. Les nouveaux moments décalés $\tilde{\mu}_{pq}^{sh}$ calculés à partir de $h(x, y)$ sont :

$$\begin{aligned} \tilde{\mu}_{pq}^{sh} &= \iint_R (x - x_g + x_{sh})^p (y - y_g + y_{sh})^q f\left(\frac{x}{b}, \frac{y}{b}\right) dx dy \\ &= \iint_R (bx - bx_g + bx_{sh})^p (by - by_g + by_{sh})^q b^2 f(x, y) dx dy \\ &= b^{p+q+2} \mu_{pq}^{sh} \end{aligned} \quad (2.58)$$

En combinant les équations (2.58) et (2.56), il est évident de voir que les $\tilde{\eta}_{pq}^{sh}$ d'une image ayant subi un changement d'échelle est la même qu'une image originale. η_{pq}^{sh} est donc invariant aux changements d'échelle.

2.2.2.5 Les moments 2D "shifted" pour une image ayant subie une rotation

Dans cette partie, deux techniques sont proposées pour obtenir l'invariance à la rotation pour les nouveaux moments. La première méthode consiste à faire une rotation de l'image en utilisant la méthode des axes principaux [Hu62] et les moments obtenus sont invariants à la rotation. Ceci est effectué en calculant les moments centraux de second ordre en utilisant la relation (2.19) et l'obtention de

l'angle θ selon l'équation (2.18). Les nouveaux moments sont calculés à l'aide de la relation (2.50) pour les images continues.

Dans la seconde technique, pour obtenir l'invariance à la rotation pour les nouveaux moments, nous utilisons la relation (2.56). Si l'image subie une rotation d'angle θ , les moments changent selon la relation

$$\mu_{pq}^r = \sum_{i=0}^p \sum_{j=0}^q \binom{p}{i} \binom{q}{j} r_{11}^i r_{21}^j r_{12}^{p-i} r_{22}^{q-j} \mu_{i+j, p+q-(i+j)} \quad (2.59)$$

où r_{mn} sont les éléments de la matrice de rotation dans le plan image d'angle ($\alpha = \theta - \theta_r$) :

$$\begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \quad (2.60)$$

θ_r est la valeur fixe de l'orientation de l'image et l'angle θ est obtenu à partir de la relation (2.18) et μ_{pq} est calculé à partir de (2.50).

En utilisant la loi binomiale, nous pouvons établir une relation entre les moments centrés et les moments "shifted". En développant la relation (2.50), on obtient :

$$\begin{aligned} \mu_{pq}^{sh} &= \iint_R \left[\sum_{i=0}^p \binom{p}{i} (x - x_g)^{p-i} x_{sh}^i \sum_{j=0}^q \binom{q}{j} (y - y_g)^{q-j} y_{sh}^j \right] dx dy \\ &= \sum_{i=0}^p \sum_{j=0}^q \binom{p}{i} \binom{q}{j} x_{sh}^i y_{sh}^j \mu_{p-i, q-j} \end{aligned} \quad (2.61)$$

La relation (2.61) ne s'applique que dans le cas des images continues. En combinant l'équation (2.61) et (2.59), nous obtenons les moments μ_{pq}^{shr} qui sont invariants aux mouvements de rotations. En utilisant cette méthode, nous donnons ci-dessous les équations des moments invariants μ_{pq}^{shr} jusqu'à l'ordre 3 :

$$\begin{aligned} \mu_{20}^{shr} &= \mu_{20}^r + m_{00} x_{sh}^2 \\ \mu_{02}^{shr} &= \mu_{02}^r + m_{00} y_{sh}^2 \\ \mu_{11}^{shr} &= \mu_{11}^r + m_{00} x_{sh} y_{sh} \\ \mu_{30}^{shr} &= \mu_{30}^r + 3\mu_{20}^r x_{sh} + m_{00} x_{sh}^3 \\ \mu_{03}^{shr} &= \mu_{03}^r + 3\mu_{02}^r y_{sh} + m_{00} y_{sh}^3 \\ \mu_{21}^{shr} &= \mu_{21}^r + \mu_{20}^r y_{sh} + 2\mu_{11}^r x_{sh} + m_{00} x_{sh}^2 y_{sh} \\ \mu_{12}^{shr} &= \mu_{12}^r + \mu_{02}^r x_{sh} + 2\mu_{11}^r y_{sh} + m_{00} y_{sh}^2 x_{sh} \end{aligned}$$

Où les paramètres "shifted" x_{sh} et y_{sh} sont obtenus à partir de l'équation (2.51) ou (2.52). Les moments μ_{pq}^{shr} sont ensuite insérés dans l'équation (2.56) pour obtenir η_{pq}^{sh} qui sont invariants aux mouvement de translation, d'échelle et de rotation.

2.2.2.6 Matrices d'interaction associées aux moments 2D "shifted"

Considérons les moments décalés définis par la relation (2.50) et considérons aussi la relation liant les moments centrés et les moments décalés définis par la relation (2.61); de (2.14) et (2.61), la forme générale de la matrice d'interaction relative aux moments décalés peut s'écrire :

$$\mathbf{L}_{\mu_{pq}^{sh}} = \sum_{i=0}^p \sum_{j=0}^q \binom{p}{i} \binom{q}{j} \left[\left(i x_{sh}^{i-1} y_{sh}^j \mathbf{L}_{x_{sh}} + j x_{sh}^i y_{sh}^{j-1} \mathbf{L}_{y_{sh}} \right) \mu_{p-i, q-j} + x_{sh}^i y_{sh}^j \mathbf{L}_{\mu_{p-i, q-j}} \right] \quad (2.62)$$

Les matrices d'interactions correspondantes à x_{sh} , y_{sh} et "a" peuvent s'écrire en prenant les dérivées de (2.51), (2.52) et (2.53) :

- En prenant le point \mathbf{P}_1

$$\begin{cases} \mathbf{L}_{x_{sh1}} = \mathbf{L}_{x_g} + \mathbf{L}_a \cos \theta - a \mathbf{L}_\theta \sin \theta \\ \mathbf{L}_{y_{sh1}} = \mathbf{L}_{y_g} + \mathbf{L}_a \sin \theta + a \mathbf{L}_\theta \cos \theta \\ \mathbf{L}_a = \frac{1}{4} (\mu_{20} + \mu_{02})^{-\frac{3}{4}} (\mathbf{L}_{\mu_{20}} + \mathbf{L}_{\mu_{02}}) \end{cases} \quad (2.63)$$

- En prenant le point \mathbf{P}_2

$$\begin{cases} \mathbf{L}_{x_{sh2}} = \mathbf{L}_a \cos \left(\theta + \frac{\pi}{2} \right) - a \mathbf{L}_\theta \sin \left(\theta + \frac{\pi}{2} \right) \\ \mathbf{L}_{y_{sh2}} = \mathbf{L}_a \sin \left(\theta + \frac{\pi}{2} \right) + a \mathbf{L}_\theta \cos \left(\theta + \frac{\pi}{2} \right) \\ \mathbf{L}_a = \frac{1}{4} (\mu_{20} + \mu_{02})^{-\frac{3}{4}} (\mathbf{L}_{\mu_{20}} + \mathbf{L}_{\mu_{02}}) \end{cases} \quad (2.64)$$

$\mathbf{L}_{\mu_{pq}}$ est la matrice d'interaction relative à μ_{pq} qui s'obtient en utilisant la relation (2.50).

Nous rappelons que

$$\theta = \frac{1}{2} \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (2.65)$$

De [TC05b] on en déduit la matrice d'interaction relative à l'angle de rotation θ :

$$\mathbf{L}_\theta = [(\mu_{20} - \mu_{02}) \mathbf{L}_{\mu_{11}} - \mu_{11} (\mathbf{L}_{\mu_{20}} - \mathbf{L}_{\mu_{02}})] / \Delta \quad (2.66)$$

où

$$\Delta = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \quad (2.67)$$

En développant l'équation (2.66), la matrice d'interaction relative à l'angle de rotation θ s'écrit :

$$\mathbf{L}_\theta = (\theta_{v_x} \theta_{v_y} \theta_{v_z} \theta_{w_x} \theta_{w_y} - 1) \quad (2.68)$$

où

$$\begin{cases} \theta_{v_x} = -aA + bB \\ \theta_{v_y} = -cA - aB \\ \theta_{v_z} = -A\theta_{w_y} + B\theta_{w_x} \\ \theta_{w_x} = -bx_g + ay_g + d \\ \theta_{w_y} = ax_g - cy_g + e \end{cases} \quad \text{et} \quad \begin{cases} a = \mu_{11}(\mu_{20} + \mu_{02})/\Delta \\ b = (2\mu_{11}^2 + \mu_{02}(\mu_{02} - \mu_{20}))/\Delta \\ c = (2\mu_{11}^2 + \mu_{20}(\mu_{20} - \mu_{02}))/\Delta \\ d = 5(\mu_{12}(\mu_{20} - \mu_{02}) + \mu_{11}(\mu_{03} - \mu_{21}))/\Delta \\ e = 5(\mu_{21}(\mu_{02} - \mu_{20}) + \mu_{11}(\mu_{30} - \mu_{12}))/\Delta \end{cases}$$

2.2.3 Choix du vecteur d'informations visuelles \mathbf{s}

Différents vecteurs d'informations visuelles \mathbf{s} ont été construits sur la base des moments de l'image pour la réalisation de tâches référencées vision [CH01], [Cha04], [TC05a], [MC07], [MKC10]. Dans la suite nous proposons une large amélioration de cette méthode qui utilise les moments de l'image comme informations visuelles.

La recherche du vecteur d'informations visuelles permet de déterminer les primitives qui permettront au système d'avoir un comportement optimal. Pour cela, il faut s'assurer que la matrice d'interaction ne soit pas singulière. Les informations visuelles choisies doivent donc permettre d'obtenir une matrice d'interaction \mathbf{L}_s de rang plein 6 dans un domaine le plus large possible. Pour assurer cette condition, il suffira de réaliser un schéma de commande le plus découplé possible, ie pouvoir associer à chaque degré de liberté à commander, une caractéristique extraite de l'image qui ne dépendra que de ce degré de liberté. On souhaite donc avoir une matrice d'interaction diagonale. Malheureusement, ce découplage est idéal et très difficile à atteindre. Nous allons dans ce qui suit, faire un découplage par bloc afin d'obtenir une matrice d'interaction triangulaire par bloc qui découplerait les mouvements de translations et de rotations. Un tel découplage nous permettra par la suite d'éviter facilement les singularités et de pouvoir déterminer si possible la courbe de singularité.

Dans cette partie, nous ferons un état de l'art sur les informations visuelles utilisant les moments invariants, ensuite nous proposerons une méthode plus efficace basée sur les moments "shifted".

2.2.3.1 Informations visuelles pour commander les translations

En asservissement visuel 2D, le comportement des primitives dans l'image est généralement satisfaisant. Cependant, la trajectoire du robot est imprévisible et peu satisfaisante pour de grands déplacements en rotation [Cha98]. En fait, la

différence entre le comportement dans l'image et l'espace 3D est due aux non-linéarités dans la matrice d'interaction. Notre objectif est d'obtenir une matrice d'interaction diagonale qui varie très peu autour de la position désirée. Les informations visuelles présentées dans [Tah04] sont basées sur les invariants d'une projection plane pour obtenir une matrice triangulaire si l'objet est parallèle au plan de l'image. En outre pour ces positions les éléments de la matrice d'interaction qui correspondent aux translations forment un bloc diagonal constant. Pour la suite des calculs, nous supposons que la position désirée au plan image est parallèle à l'objet (ie $A=B=0$). Et on note \mathbf{L}_s^{\parallel} la matrice d'interaction correspondante.

Classiquement, un choix convenable consiste à sélectionner au moins les moments :

- d'ordre 0 : m_{00} , c'est-à-dire l'aire de la région \mathbf{R} de l'image ;
- d'ordre 1 : m_{10} et m_{01} , qui sont liés aux coordonnées du centre de gravité de \mathbf{R} : $x_g = \frac{m_{10}}{m_{00}}$ et $y_g = \frac{m_{01}}{m_{00}}$. La matrice d'interaction correspondante est alors facilement calculable à partir de la forme générale (2.47). On obtient alors [CH01, Cha02] :

$$\begin{bmatrix} \mathbf{L}_{x_g}^{\parallel} \\ \mathbf{L}_{y_g}^{\parallel} \\ \mathbf{L}_a^{\parallel} \end{bmatrix} = \begin{bmatrix} -C & 0 & Cx_g & \varepsilon_1 & -(1 + \varepsilon_2) & y_g \\ 0 & -C & Cy_g & 1 + \varepsilon_3 & -\varepsilon_1 & -x_g \\ 0 & 0 & 2aC & 3ay_g & -3ax_g & 0 \end{bmatrix} \quad (2.69)$$

avec $\varepsilon_1 = 4(\eta_{11} + x_g y_g)$, $\varepsilon_2 = 4(\eta_{20} + x_g^2)$ et $\varepsilon_3 = 4(\eta_{02} + y_g^2)$

On remarque que cette matrice est bien triangulaire mais toute fois, on note la forte linéarité de ses éléments. On n'a aussi une même dynamique sur chaque axe de translation. Une autre façon de faire serait de choisir

$$a_n = Z^* \sqrt{\frac{a^*}{a}}, \quad x_n = a_n x_g, \quad y_n = a_n y_g \quad (2.70)$$

Où a^* est la surface désirée de l'objet dans l'image et où Z^* est la profondeur désirée entre la caméra et l'objet. La matrice d'interaction associée à ces informations visuelles normalisées peut donc s'écrire à partir de la relation (2.47) [Tah04] :

$$\begin{aligned} \mathbf{L}_{x_n}^{\parallel} &= \begin{bmatrix} -1 & 0 & 0 & a_n \varepsilon_{11} & -a_n (1 + \varepsilon_{12}) & y_n \end{bmatrix} \\ \mathbf{L}_{y_n}^{\parallel} &= \begin{bmatrix} 0 & -1 & 0 & a_n (1 + \varepsilon_{21}) & -a_n \varepsilon_{11} & -x_n \end{bmatrix} \\ \mathbf{L}_{a_n}^{\parallel} &= \begin{bmatrix} 0 & 0 & -1 & -3y_n/2 & 3x_n/2 & 0 \end{bmatrix} \end{aligned} \quad (2.71)$$

où $\varepsilon_{11} = n_{11} - x_g y_g / 2$, $\varepsilon_{12} = n_{20} - x_g^2 / 2$ et $\varepsilon_{21} = n_{02} - y_g^2 / 2$

Le choix de x_n et y_n permet un découplage total des 3 degrés de liberté de translation. On obtient aussi la même dynamique sur les trois axes de translation.

Notons le bloc diagonal égal à \mathbf{I}_3 dans l'équation (2.71). Cette bonne propriété permettra d'obtenir une bonne trajectoire des translations de la caméra.

2.2.3.2 Informations visuelles pour commander les rotations

L'angle θ défini par l'équation (2.65) est généralement choisi pour contrôler l'orientation autour de l'axe optique. A partir de (2.68) on a :

$$\mathbf{L}_\theta^\parallel = \begin{bmatrix} 0 & 0 & 0 & \theta_{w_x} & \theta_{w_y} & -1 \end{bmatrix} \quad (2.72)$$

où

$$\begin{cases} \theta_{w_x} = -bx_g + ay_g + d \\ \theta_{w_y} = ax_g - cy_g + e \end{cases} \quad (2.73)$$

et

$$\begin{cases} a = \mu_{11}(\mu_{20} + \mu_{02})/\Delta \\ b = (2\mu_{11}^2 + \mu_{02}(\mu_{02} - \mu_{20}))/\Delta \\ c = (2\mu_{11}^2 + \mu_{20}(\mu_{20} - \mu_{02}))/\Delta \\ d = 5(\mu_{12}(\mu_{20} - \mu_{02}) + \mu_{11}(\mu_{03} - \mu_{21}))/\Delta \\ e = 5(\mu_{21}(\mu_{02} - \mu_{20}) + \mu_{11}(\mu_{30} - \mu_{12}))/\Delta \\ \Delta = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \end{cases}$$

On rappelle que l'angle θ dépend essentiellement de la rotation w_z autour de l'axe optique. Pour contrôler les deux derniers mouvements de rotation, à savoir les rotations autour de l'axe (w_x) et (w_y), des combinaisons des moments invariants ont été choisies parmi celles données ci-dessous [Tah04] :

$$\begin{aligned} r_1 &= \frac{I_{n1}}{I_{n2}}, r_2 = \frac{I_{n3}}{I_{n4}}, r_3 = \frac{I_{n5}}{I_{n6}}, r_4 = \frac{I_{n7}}{I_{n6}}, r_5 = \frac{I_{n8}}{I_{n6}}, \\ r_6 &= \frac{I_{n9}}{I_{n6}}, r_7 = \frac{I_{n11}}{I_{n10}}, r_8 = \frac{I_{n12}}{I_{n10}}, r_9 = \frac{I_{n13}}{I_{n15}}, r_{10} = \frac{I_{n14}}{I_{n15}} \end{aligned} \quad (2.74)$$

$$I_{n1} = -\mu_{20}\mu_{02} + \mu_{11}^2$$

$$I_{n2} = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$$

$$I_{n3} = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})(3\mu_{21} - \mu_{03})^2$$

$$I_{n4} = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$$

$$I_{n5} = -\mu_{30}^2\mu_{03}^2 + 6\mu_{30}\mu_{21}^3 - 4\mu_{30}\mu_{12}^3 - 4\mu_{03}\mu_{21}^3 + 3\mu_{21}^2\mu_{12}^2$$

$$I_{n6} = 3\mu_{30}^2\mu_{12}^2 + 2\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}^3\mu_{12} - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 2\mu_{30}\mu_{12}^3 + 3\mu_{21}^4 \\ + 2\mu_{03}\mu_{21}^3 + 3\mu_{21}^2\mu_{03}^2 - 6\mu_{21}\mu_{12}^2\mu_{03} + 3\mu_{12}^4 - 6\mu_{21}\mu_{12}^2\mu_{03} + 3\mu_{12}^4$$

$$I_{n7} = -\mu_{03}\mu_{30}^3 + 3\mu_{30}^2\mu_{21}\mu_{12} - 2\mu_{30}\mu_{21}^3 - 3\mu_{30}\mu_{21}^2\mu_{03} + 6\mu_{30}\mu_{21}\mu_{12}^2 \\ + 3\mu_{12}^2\mu_{30}\mu_{03} + \mu_{30}\mu_{03}^3 - 3\mu_{12}\mu_{21}^3 - 6\mu_{03}\mu_{12}\mu_{21}^2 + 3\mu_{21}\mu_{12}^3 \\ - 3\mu_{03}^2\mu_{21}\mu_{12} + 2\mu_{03}\mu_{12}^3$$

$$I_{n8} = -\mu_{12}\mu_{30}^3 + \mu_{30}^2\mu_{21}^2 - \mu_{30}^2\mu_{21}\mu_{03} - 2\mu_{30}^2\mu_{12}^2 + 3\mu_{30}\mu_{21}^2\mu_{12} \\ - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 3\mu_{30}\mu_{12}^3 - \mu_{03}^2\mu_{30}\mu_{12} + 3\mu_{03}\mu_{21}^3 + 3\mu_{03}\mu_{21}^3 \\ - 2\mu_{03}^2\mu_{21}^2 + 3\mu_{12}^2\mu_{03}\mu_{21} - \mu_{21}\mu_{03}^3 + \mu_{12}^2\mu_{03}^2$$

$$I_{n9} = \mu_{30}^4 + 6\mu_{12}\mu_{30}^3 + 6\mu_{03}\mu_{30}^2\mu_{21} + 9\mu_{30}^2\mu_{12}^2 + 2\mu_{30}^2\mu_{03}^2 + 18\mu_{30}\mu_{21}\mu_{12}\mu_{03} \\ + 6\mu_{30}\mu_{12}\mu_{03}^2 + 9\mu_{21}^2\mu_{03}^2 + 6\mu_{21}\mu_{03}^3 + \mu_{03}^4$$

$$I_{n10} = \mu_{40}\mu_{04} - 4\mu_{31}\mu_{13} + 3\mu_{22}\mu_{22}$$

$$I_{n11} = 3\mu_{40}\mu_{22} - 2\mu_{40}\mu_{04} + 3\mu_{31}\mu_{31} + 2\mu_{31}\mu_{13} - 3\mu_{22}\mu_{04} + 3\mu_{13}\mu_{13}$$

$$I_{n12} = 3\mu_{40}\mu_{40} + 12\mu_{40}\mu_{22} + 2\mu_{40}\mu_{04} + 16\mu_{31}\mu_{13} + 12\mu_{22}\mu_{04} + 3\mu_{04}\mu_{04}$$

$$I_{n13} = (\mu_{50} + 2\mu_{32} + \mu_{14})^2 + (\mu_{05} + 2\mu_{23} + \mu_{41})^2$$

$$I_{n14} = (\mu_{50} - 2\mu_{32} - 3\mu_{14})^2 + (\mu_{05} - 2\mu_{23} - 3\mu_{41})^2$$

$$I_{n15} = (\mu_{50} - 10\mu_{32} + 5\mu_{14})^2 + (\mu_{05} - 10\mu_{23} + 5\mu_{41})^2$$

Ces combinaisons sont invariantes aux mouvements de translation parallèles au plan image, aux changements d'échelles (aux translations selon l'axe optique si l'objet est parallèle au plan image), ainsi qu'au mouvement de rotation autour de l'axe optique. La matrice d'interaction liée aux invariants (2.74) peut être obtenue à partir de (2.49). On obtient alors :

$$\begin{aligned} \mathbf{L}_{ri}^{\parallel} &= \begin{bmatrix} 0 & 0 & 0 & r_{iwx} & r_{iwy} & 0 \end{bmatrix} \\ \mathbf{L}_{rj}^{\parallel} &= \begin{bmatrix} 0 & 0 & 0 & r_{jwx} & r_{jwy} & 0 \end{bmatrix} \end{aligned} \quad (2.75)$$

Où les termes r_{iwx} , r_{iwy} , r_{jwx} et r_{jwy} ne sont pas donnés ici. Comme prévu, on peut noter l'invariance de ces informations visuelles par rapport à tout mouvement de translation 3D et de rotation par rapport à w_z lorsque bien sûr la position désirée au plan image est parallèle à l'objet. Nous précisons que le choix des invariants dépend de la forme de l'objet par rapport auquel on veut se positionner. Ces invariants sont choisis tels que leurs matrices d'interaction associées $\mathbf{L}_{r_i}^{\parallel}$ et $\mathbf{L}_{r_j}^{\parallel}$ soient les plus orthogonales possible et présentent un domaine de convergence le plus large possible.

Enfin, dans le cas d'un objet symétrique (un rectangle par exemple), les moments centrés d'ordre impair deviennent nuls pour toutes les positions où l'image de l'objet est également symétrique. L'utilisation de la plupart de ces invariants pour contrôler les vitesses de rotation w_x et w_y est donc impossible. Pour ce genre d'objet, un découplage a été réalisé dans [Cha02] en utilisant les informations visuelles suivantes :

$$\begin{cases} s_x = \frac{c_2 c_3 + s_2 s_3}{K} \\ s_y = \frac{s_2 c_3 - c_2 s_3}{K} \end{cases} \begin{cases} c_3 = c_1^2 - s_1^2 \\ s_3 = 2s_1 c_1 \\ K = \frac{H_1 H_2^{(3/2)}}{\sqrt{a}} \end{cases} \quad (2.76)$$

$$\begin{cases} H_1 = c_1^2 + s_1^2 \\ H_2 = \mu_{20} + \mu_{02} \end{cases} \quad \text{où} \quad \begin{cases} c_1 = \mu_{20} - \mu_{02} \quad , \quad s_1 = 2\mu_{11} \\ c_2 = \mu_{03} - 3\mu_{21} \quad , \quad s_1 = \mu_{30} - 3\mu_{12} \end{cases}$$

2.2.3.3 Nouvelles informations visuelles

Après cet état de l'art sur les informations visuelles existantes utilisant les moments de l'image, nous proposerons dans ce paragraphe de nouvelles informations visuelles utilisant les moments dont le centre de gravité a été décalé. Nous rappelons les six informations visuelles basées sur les moments pour contrôler les six degrés de liberté de la caméra :

$$\mathbf{s} = (x_n \quad y_n \quad a_n \quad r_i \quad r_j \quad \theta) \quad (2.77)$$

Nous rappelons aussi que la matrice d'interaction associée à ces six informations visuelles pour une position de la caméra telle que son plan image soit parallèle à l'objet a la forme suivante :

$$\begin{aligned} \mathbf{L}_{x_n}^{\parallel} &= \begin{bmatrix} -1 & 0 & 0 & a_n \varepsilon_{11} & -a_n (1 + \varepsilon_{12}) & y_n \end{bmatrix} \\ \mathbf{L}_{y_n}^{\parallel} &= \begin{bmatrix} 0 & -1 & 0 & a_n (1 + \varepsilon_{21}) & -a_n \varepsilon_{11} & x_n \end{bmatrix} \\ \mathbf{L}_{a_n}^{\parallel} &= \begin{bmatrix} 0 & 0 & -1 & -3y_n/2 & 3x_n/2 & 0 \end{bmatrix} \\ \mathbf{L}_{r_i}^{\parallel} &= \begin{bmatrix} 0 & 0 & 0 & r_{iwx} & r_{iwy} & 0 \end{bmatrix} \\ \mathbf{L}_{r_j}^{\parallel} &= \begin{bmatrix} 0 & 0 & 0 & r_{jwx} & r_{jwy} & 0 \end{bmatrix} \\ \mathbf{L}_{\theta}^{\parallel} &= \begin{bmatrix} 0 & 0 & 0 & \theta_{w_x} & \theta_{w_y} & -1 \end{bmatrix} \end{aligned}$$

On rappelle de nouveau le bon découplage entre les degrés de liberté de la caméra en utilisant ces combinaisons de moments. Si la matrice $\mathbf{L}_{rij}^{\parallel}$ correspondant au mouvement de rotation le long de l'axe x et l'axe y est non singulière :

$$\mathbf{L}_{rij}^{\parallel} = \begin{bmatrix} r_{iwx} & r_{iwy} \\ r_{jwx} & r_{jwy} \end{bmatrix} \quad (2.78)$$

Alors la matrice d'interaction relative aux informations visuelles

$\mathbf{s} = (x_n \ y_n \ a_n \ r_i \ r_j \ \theta)$ sera non singulière. Malheureusement, le choix des couples d'invariants (r_i, r_j) à introduire dans la boucle de commande dépend de l'objet considéré. Pour différents type d'objet, il faudra choisir le meilleur couple d'invariants (r_i, r_j) qui permet d'avoir à la fois un bon conditionnement de la matrice d'interaction et un bon comportement de l'erreur $\mathbf{e}_r = (r_i - r_i^*)^2 + (r_j - r_j^*)^2$ en fonction des rotations w_x et w_y autour de la position désirée. Ce choix est d'autant plus difficile lorsque l'objet est de forme symétrique. De plus, à l'analyse des invariants, nous pouvons constater que les moments d'ordre trois à cinq ont été utilisés. Toutefois, les fonctions $x^p y^q$ intervenant dans les moments ne sont pas orthogonales. De ce fait, bien que l'usage de moments d'ordres élevés permette de caractériser des détails de plus en plus précis, la dynamique de leurs valeurs tend à s'amoinrir quelle que soit la forme. Teh et Chin montrent dans [TC88] que les moments invariants ne sont pas des descripteurs robustes en présence de bruit.

Au vue de cette analyse critique, nous pouvons donc dire que l'étude de la stabilité en asservissement visuel reste encore un champ ouvert pour la recherche.

Notre contribution dans ce travail est de trouver un unique couple (r_i, r_j) comme informations visuelles pour contrôler les rotations autour de l'axe w_x et w_y . Ce couple d'invariants ne devrait pas dépendre de la forme de l'objet. Pour réduire le bruit dans la mesure, les moments d'ordre faible seront utilisés. Le couple (r_i, r_j) ainsi choisi, devrait être calculé en utilisant les moments 2D extraits de l'image dont le centre de gravité (x_g, y_g) aurait subi un décalage. Le nouveau centre de gravité $(x_g + x_{sh}, y_g + y_{sh})$ respecte toutes les propriétés d'invariance à savoir l'invariance aux mouvements de translation, d'échelle et de rotation de la caméra.

Considérons les trois polynômes suivants :

$$I_{s1} = -\mu_{30}^{sh}\mu_{12}^{sh} + \mu_{21}^{sh}\mu_{21}^{sh} - \mu_{03}^{sh}\mu_{21}^{sh} + \mu_{12}^{sh}\mu_{12}^{sh} \quad (2.79)$$

$$I_{s2} = \mu_{20}^{sh}\mu_{02}^{sh} - \mu_{11}^{sh}\mu_{11}^{sh} \quad (2.80)$$

$$I_{s3} = 3\mu_{30}^{sh}\mu_{12}^{sh} + \mu_{30}^{sh}\mu_{30}^{sh} + 3\mu_{03}^{sh}\mu_{21}^{sh} + \mu_{03}^{sh}\mu_{03}^{sh} \quad (2.81)$$

Où les μ_{pq}^{sh} sont calculés en utilisant la relation (2.61) que nous rappelons ici :

$$\begin{aligned}
\mu_{pq}^{sh} &= \iint_R \left[\sum_{i=0}^p \binom{p}{i} (x - x_g)^{p-i} x_{sh}^i \sum_{j=0}^q \binom{q}{j} (y - y_g)^{q-j} y_{sh}^j \right] dx dy \\
&= \sum_{i=0}^p \sum_{j=0}^q \binom{p}{i} \binom{q}{j} x_{sh}^i y_{sh}^j \mu_{p-i, q-j}
\end{aligned} \tag{2.82}$$

Si les paramètres de décalage sont nuls, alors

$$r_{s1} = \frac{I_{s1}}{I_{s2}/10}$$

ou

$$r_{s2} = \frac{I_{s3}}{I_{s2}/10}$$

ou

$$r_{s3} = \frac{I_{s3}}{I_{s1}}$$

sont invariants aux mouvements de translation, d'échelle et de rotation. Dans cette optique, les points \mathbf{P}_1 et \mathbf{P}_2 doivent être choisis de telle sorte que r_{s1} , r_{s2} et r_{s3} , restent invariants aux mouvements de translation, d'échelle et de rotation dans le plan image. Pour cela, les points \mathbf{P}_1 et \mathbf{P}_2 sont choisis de telle sorte qu'ils doivent subir les mêmes translations, les mêmes rotations et les mêmes changements d'échelle que l'objet dans le plan image. Pour toutes ces raisons évoquées, les points \mathbf{P}_1 et \mathbf{P}_2 doivent appartenir au contour de l'objet projeté dans le plan image. Toutefois, choisir un point qui respectera ces conditions nécessitera un traitement d'image supplémentaire pour faire l'appariement ou le "tracking" durant l'asservissement visuel. Pour éviter cela, nous pouvons exploiter le même invariant " r_{si} " dont la forme dépendra du choix des points \mathbf{P}_1 et \mathbf{P}_2 . Au regard de la figure 2.3, le point \mathbf{P}_1 peut être choisi dans la direction du vecteur \mathbf{n} et ayant comme coordonnées x_{sh1} et y_{sh1} que nous rappelons ici :

$$x_{sh1} = x_g + a \cos \theta$$

et

$$y_{sh1} = y_g + a \sin \theta$$

où le vecteur \mathbf{n} définit la direction de l'axe principale de l'objet comme le montre la figure 2.3. Si le paramètre " a " est invariant aux rotations, le point \mathbf{P}_1 subira la même rotation dans le plan image que l'objet. Si de plus le paramètre " a " subit le même mouvement d'échelle que l'objet dans le plan image, alors le point \mathbf{P}_1 subira le même mouvement d'échelle que tous les points appartenant au contour de l'objet. Dans cette situation, les invariants r_{s1} , r_{s2} et r_{s3} seront invariants aux mouvements de translation, de rotation et de changement d'échelle. Un

second point \mathbf{P}_2 peut aussi être choisi dans la direction d'un vecteur \mathbf{v} définissant l'axe principal mineur et ayant pour coordonnées x_{sh2} et y_{sh2} que nous rappelons ici :

$$x_{sh2} = x_g + a \cos \left(\theta + \frac{\pi}{2} \right)$$

et

$$y_{sh2} = y_g + a \sin \left(\theta + \frac{\pi}{2} \right)$$

Comme paramètre "a" commun aux deux points \mathbf{P}_1 et \mathbf{P}_2 nous pouvons choisir

$$a = (\mu_{20} + \mu_{02})^{\frac{1}{4}}$$

Ou bien des valeurs différentes pour chaque point en choisissant les paramètres des axes principaux

$$\begin{cases} a_1^2 = \frac{2}{\mu_{00}} \left(\mu_{02} + \mu_{20} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \right) \\ a_2^2 = \frac{2}{\mu_{00}} \left(\mu_{02} + \mu_{20} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \right) \end{cases}$$

Pour contrôler la rotation autour de l'axe w_x et w_y respectivement, les moments "shifted" ayant subi un décalage du centre de gravité μ_{pq}^{sh} dans l'équation (2.50) sont calculés en utilisant les coordonnées du point \mathbf{P}_1 et \mathbf{P}_2 respectivement. Ainsi la matrice d'interaction relative à l'un des invariants r_{s1} , r_{s2} et r_{s3} suivant l'axe w_x et w_y aura la forme

$$\begin{aligned} \mathbf{L}_{rsi}^{\parallel} &= \begin{bmatrix} 0 & 0 & 0 & r_{iwxsh1/P1} & r_{iwysh1/P1} & 0 \end{bmatrix} \\ \mathbf{L}_{rsj}^{\parallel} &= \begin{bmatrix} 0 & 0 & 0 & r_{jwxsh2/P2} & r_{jwysh2/P2} & 0 \end{bmatrix} \end{aligned} \quad (2.83)$$

Alors la matrice d'interaction relative aux nouvelles informations visuelles $\mathbf{s}_n = (x_n \ y_n \ a_n \ r_{i/P1} \ r_{j/P2} \ \theta)$ aura donc la forme suivante :

$$\begin{aligned} \mathbf{L}_{x_n}^{\parallel} &= \begin{bmatrix} -1 & 0 & 0 & a_n \varepsilon_{11} & -a_n (1 + \varepsilon_{12}) & y_n \end{bmatrix} \\ \mathbf{L}_{y_n}^{\parallel} &= \begin{bmatrix} 0 & -1 & 0 & a_n (1 + \varepsilon_{21}) & -a_n \varepsilon_{11} & x_n \end{bmatrix} \\ \mathbf{L}_{a_n}^{\parallel} &= \begin{bmatrix} 0 & 0 & -1 & -3y_n/2 & 3x_n/2 & 0 \end{bmatrix} \\ \mathbf{L}_{rsi}^{\parallel} &= \begin{bmatrix} 0 & 0 & 0 & r_{iwxsh1/P1} & r_{iwysh1/P1} & 0 \end{bmatrix} \\ \mathbf{L}_{rsj}^{\parallel} &= \begin{bmatrix} 0 & 0 & 0 & r_{jwxsh2/P2} & r_{jwysh2/P2} & 0 \end{bmatrix} \\ \mathbf{L}_{\theta}^{\parallel} &= \begin{bmatrix} 0 & 0 & 0 & \theta_{w_x} & \theta_{w_y} & -1 \end{bmatrix} \end{aligned} \quad (2.84)$$

Où les expressions $r_{iwxsh1/P1}$, $r_{iwysh1/P1}$, $r_{jwxsh2/P2}$ et $r_{jwysh2/P2}$ seront données dans le chapitre relatif à l'étude de la stabilité.

2.2.3.4 Conclusion

Dans ce chapitre, nous avons rappelé des notions sur la commande par asservissement visuel et en particulier la notion de matrice d'interaction. Par ailleurs, nous avons donné les formules analytiques de ces matrices dans le cas des moments 2D extraits de l'image. Nous avons présenté également les problèmes potentiels rencontrés en asservissement visuel. Pour remédier à ces problèmes, nous avons proposé six informations visuelles calculées à partir des moments 2D extraits de l'image d'un objet plan continu, dont le centre de gravité a été décalé. Ce choix d'informations visuelles a permis d'obtenir une matrice d'interaction creuse et de minimiser les non linéarités de celle-ci pour une position de la caméra où son plan image est parallèle à l'objet. Dans le chapitre suivant, nous appliquons ces résultats au problème de stabilité.

Chapitre 3

Etude de la stabilité

3.1 Introduction

La notion de stabilité constitue une problématique centrale de la théorie du contrôle. Différents types de problèmes de stabilité peuvent être rencontrés dans l'étude des systèmes dynamiques. Dans cette partie, nous entendons par stabilité des points d'équilibre. La stabilité d'un point d'équilibre est généralement étudiée à l'aide du concept de stabilité au sens de Lyapunov. Par définition, si un système est dans un état d'équilibre, il restera dans cet état pour t variant dans le temps. L'étude de la stabilité au sens de Lyapunov consiste en l'étude des trajectoires du système quand l'état initial est voisin d'un état d'équilibre. La stabilité au sens de Lyapunov ou la stabilité asymptotique se démontrent grâce à une "fonction de Lyapunov", qui est une généralisation de l'énergie d'un système. L'avantage de cette généralisation est que l'existence d'une fonction de Lyapunov ayant les propriétés requises est une condition nécessaire et suffisante de stabilité à la Lyapunov ou asymptotique ; son inconvénient est que la recherche d'une telle fonction est un problème dont la solution, sauf cas particulier, n'a rien d'évident. Dans le cas de systèmes linéaires aux paramètres incertains, cette recherche peut se formaliser en un problème d'optimisation et, lorsque celui-ci est convexe, il existe des algorithmes de résolution efficaces. Il existe également des méthodes permettant de réaliser un bouclage de manière qu'une fonction de Lyapunov, choisie à l'avance, garantisse la stabilité. La théorie du contrôle, au carrefour des mathématiques et de l'automatique, s'efforce d'apporter des résultats et des méthodes permettant de comprendre, d'analyser et de résoudre les problèmes associés à des systèmes contrôlés. Ces systèmes ont des variables qui permettent d'influencer sa dynamique et qui peuvent être ajustées par un opérateur ou un processus automatisé. Les domaines d'application sont essentiellement ceux de l'automatique c'est-à-dire la synthèse de lois de commande pour des processus provenant de différents domaines applicatifs comme la robotique. L'un des objectifs de l'auto-

matique est de construire un contrôle suffisamment régulier pour être utilisable en pratique. En ce qui concerne les systèmes linéaires, de nombreux outils performants permettent de construire un tel contrôle. En effet, la stabilité au sens de Lyapunov est bien maîtrisée pour les systèmes linéaires. Par contre, la stabilité des systèmes non linéaires reste encore largement inconnue dans l'industrie, bien que des techniques de linéarisation permettent d'obtenir dans certains cas de bons résultats.

3.2 Stabilité au sens de Lyapunov

La seconde méthode de Lyapunov concerne l'analyse de la stabilité interne d'un système dynamique. Le concept de stabilité au sens de Lyapunov permet de caractériser des domaines de stabilité pour des systèmes non- linéaires. Cette notion est intimement liée à la notion de fonction de Lyapunov. La définition d'une fonction scalaire \mathbf{V} appelé fonction candidate à Lyapunov est présentée comme suit :

Définition III.1

La fonction $\mathbf{V} : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^+$ est une fonction de Lyapunov, si elle satisfait les deux conditions suivantes :

1. $\mathbf{V}(x)$ est continue et ses dérivées partielles $\dot{\mathbf{V}}_i(x) = \frac{\partial \mathbf{V}}{\partial x_i}$ existent et sont continues pour tout $i=1, \dots, n$.
2. $\mathbf{V}(x) > 0$ pour tout $x \neq 0$ et $\mathbf{V}(0) = 0$.

L'existence d'une fonction de Lyapunov contractive le long des trajectoires d'un système permet de garantir la stabilité de ce système, comme cela est explicité dans le théorème suivant :

Théorème III.1

Dans un voisinage $D \subset \mathbb{R}^n$, l'équilibre $x = 0$, $x \in D$, est :

1. Localement stable s'il existe une fonction de Lyapunov $\mathbf{V}(x)$ telle que $\dot{\mathbf{V}}_i(x) < 0$ pour tout $x \in D$;
2. Localement asymptotiquement stable s'il existe une fonction de Lyapunov $\mathbf{V} : D \rightarrow \mathbb{R}^+$ telle que $\dot{\mathbf{V}}_i(x) < 0$ pour tout $x \in D$, $x \neq 0$.

Il est important de souligner que le théorème III.1 garantit la stabilité dans un sens local, c'est-à-dire dans un voisinage de l'origine contenu dans l'ensemble D . Si la convergence des trajectoires vers l'origine est garantie quelle que soit $x(0) \in D \rightarrow \mathbb{R}^n$ alors le système est dit globalement asymptotiquement stable. Ce théorème fournit aussi une condition suffisante pour la stabilité.

3.3 Rappels sur la commande référencée capteur des robots manipulateurs

3.3.1 Approche fonction de tâche appliquée aux asservissements visuels

Le formalisme des fonctions de tâche pour la commande de robots manipulateurs qui suit est tiré de [CS91]. Le lecteur intéressé est invité à consulter cet ouvrage pour un exposé plus approfondi.

Ce formalisme permet d'exprimer une tâche robotique à effectuer comme étant la régulation vers zéro sur un horizon temporel d'une fonction $\mathbf{e}(\mathbf{q}(t), t)$ appelée fonction de tâche. Cette fonction appartient à la classe C^2 , de même dimension n que le vecteur $\mathbf{q}(t)$ de coordonnées généralisées du robot. Elle exprime l'erreur entre la configuration courante et la configuration souhaitée du capteur par rapport à la scène. Dans le cas de l'asservissement visuel 2D, le positionnement de la caméra par rapport à une cible visuelle, est caractérisée par un vecteur $\mathbf{s}(t)$ de $k > 6$ informations issues de l'image. La fonction de tâche est généralement définie sous la forme suivante :

$$\mathbf{e}(\mathbf{q}(t), t) = \mathbf{C}(\mathbf{s}(\mathbf{r}(q), t) - \mathbf{s}^*) \quad (3.1)$$

- $\mathbf{s}(\mathbf{r}(q), t)$ est le vecteur de mesures de dimension k obtenu à partir des informations visuelles courantes. Il exprime le fait que les données visuelles à l'instant courant dépendent de la situation de la caméra mais également du temps si la cible est animée d'un mouvement propre ;

- \mathbf{s}^* représente la valeur que \mathbf{s} doit atteindre pour que la tâche soit correctement réalisée. En d'autre terme c'est la consigne sur les informations visuelles de la tâche. Elle peut être constante ou variable en fonction du temps ;

- \mathbf{C} est une matrice de combinaison de dimension $n \times k$ dite matrice de combinaison, avec n correspondant au nombre de degrés de liberté du robot à commander. Elle permet de prendre en compte par combinaison plus d'informations que de degrés de liberté, de manière à gérer une éventuelle redondance de l'information ;

- $\mathbf{r}(\mathbf{q}(t), t)$ est la pose du capteur. C'est la position entre la caméra et son environnement.

Rappelons que le jacobien du robot $\mathbf{J}(\mathbf{q})$ exprime le lien différentiel entre la situation $\mathbf{r}(t)$ de la caméra et le vecteur de coordonnées généralisées $\mathbf{q}(t)$:

$$\dot{\mathbf{r}}(t) = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \quad (3.2)$$

Où $\mathbf{J}(\mathbf{q}) = \frac{\partial \mathbf{r}}{\partial \mathbf{q}}$ est la matrice jacobienne du robot exprimant le passage entre la vitesse de l'effecteur et la vitesse articulaire du robot. Nous supposons par

la suite que le jacobien du robot est parfaitement connu et inversible.

Pour poser correctement le problème de la régulation de la fonction de tâche, Samson et al. [CS91] ont introduit une condition d'admissibilité de la tâche qui est vérifiée si \mathbf{e} possède certaines propriétés que nous allons définir par la suite. L'équation (3.1) ainsi définie, la fonction de tâche apparaît comme une fonction d'erreur et le problème de commande consiste à réguler cette erreur c'est-à-dire l'amener à zéro et à l'y maintenir. La notion de ρ -admissibilité, définie dans [CS91], permet d'établir des conditions suffisantes de régularité pour que cette condition soit assurée.

Pour bien poser le problème de régulation de la fonction de tâche $\mathbf{e}(\mathbf{q}(t), t)$ comme cela est précisé dans Samson et al. [CS91], la fonction de tâche doit posséder certaines propriétés, comme l'unicité de la trajectoire $\mathbf{q}_r(t)$ qui est solution de la condition $\mathbf{e}(\mathbf{q}(t), t) = 0 \forall t \in [0, T]$. La résolution de l'équation $\mathbf{e}(\mathbf{q}(t), t) = 0$ n'étant en général pas unique à cause de la non-linéarité de la fonction de tâche, peut conduire à la définition de plusieurs trajectoires qui seront des solutions distinctes. Pour pallier à cet inconvénient, il est nécessaire d'introduire une condition initiale $\mathbf{q}_r(0) = \mathbf{q}_0$, qui vérifie $\mathbf{e}(\mathbf{q}_0, 0) = 0$. Notons qu'une trajectoire solution vérifiant cette propriété additionnelle est dite trajectoire idéale [CS91]. Cette dernière condition est encore insuffisante pour assurer la condition de régularité recherchée. Pour que le problème soit bien posé, il est nécessaire que le jacobien de la tâche $\dot{\mathbf{e}} = \frac{\partial \mathbf{e}}{\partial \mathbf{q}}$ soit non singulier. D'autre part, outre l'unicité de la trajectoire solution, il est nécessaire d'une part que la régulation de la fonction de tâche vers 0 assure la convergence de la trajectoire du robot vers la trajectoire solution désirée, et d'autre part qu'une petite variation de \mathbf{e} induit une petite variation de \mathbf{q} . Lorsque toutes les conditions requises sont satisfaites, la propriété de ρ -admissibilité offre les conditions suffisantes puisqu'elle permet d'assurer l'existence d'un difféomorphisme local \mathbf{F} entre un ensemble de couples (\mathbf{q}, t) , incluant la condition initiale \mathbf{q}_0 , et l'ensemble des couples (\mathbf{e}, t) correspondants.

Pour contrôler un robot la commande cinématique qui consiste à envoyer périodiquement des consignes de vitesses articulaires au robot, est la plus utilisée. Le système de commande considéré est donc :

$$\dot{\mathbf{q}} = \mathbf{v} \quad (3.3)$$

avec $\mathbf{q} \in \mathbf{Q}$ le vecteur de configuration articulaire du robot ($\dim(\mathbf{Q}) = n$) et $\mathbf{v} \in \mathbf{R}^n$ le vecteur des vitesses articulaires assimilé au vecteur de commande. Les moteurs utilisés dans la plupart des robots disposent d'un asservissement en vitesse de bas niveau. Ceci justifie donc le choix de la commande cinématique. L'hypothèse sous-jacente est que cet asservissement est capable d'assurer l'égalité entre la vitesse réelle des moteurs et la vitesse désirée (i.e. $\dot{\mathbf{q}} = \dot{\mathbf{q}}^*$). Cette hypothèse ne pourra être validée que pour des faibles accélérations nécessitant un découplage dynamique des articulations du robot [CS91]. Dans la théorie clas-

sique du contrôle de robots, un problème de commande est exprimé comme la régulation d'une fonction de sortie du système robotique ; ce dernier étant défini comme l'ensemble robot plus capteurs. Les fonctions de tâche constituent une classe particulière de fonctions de sortie de classe C^k , $k = 1$, à valeurs dans \mathbf{R}^n , qui dépendent du vecteur \mathbf{q} de configuration articulaire du robot et du temps en tant que paramètre indépendant. L'approche de commande par fonctions de tâche, développée dans [CS91], fournit un cadre général pour la synthèse des lois de commande pour les robots manipulateurs. Nous pouvons trouver des fonctions de tâche exprimées dans :

- l'espace articulaire du robot :

$\mathbf{e}(\mathbf{q}, t) = \mathbf{r}(\mathbf{s}(\mathbf{q})) - \mathbf{r}(\mathbf{s}^*(t))$. La position \mathbf{r} de l'effecteur du robot est estimée à partir des informations visuelles. Cette approche est appelée asservissement visuel 3D car le contrôle de la caméra s'effectue dans l'espace cartésien [WW96] et [PM99].

- l'espace euclidien :

$\mathbf{e}(\mathbf{q}, t) = \bar{\mathbf{q}}(\mathbf{q}(\mathbf{q})) - \bar{\mathbf{q}}^*(\mathbf{q}^*(t))$. La configuration articulaire du robot est observée par une caméra déportée. On parle dans ce cas d'asservissement visuel dans l'espace articulaire. La caméra qui observe le robot dispose généralement d'un grand champ de vision pour que toutes les articulations du robot soient visibles. Cette méthode a été utilisée pour un robot à un axe par [YC89] mais en utilisant au moins deux caméras.

- espace d'un capteur :

$\mathbf{e}(\mathbf{q}, t) = \mathbf{s}(\mathbf{q}, t) - \mathbf{s}^*(t)$, avec \mathbf{s} le signal du capteur. On peut encore l'écrire sous la forme :

$\mathbf{e}(\mathbf{q}, t) = \mathbf{C}(\mathbf{s}(\mathbf{q}) - \mathbf{s}^*(t))$ où \mathbf{C} est une matrice qui permet de tenir compte d'une éventuelles redondance d'information. La fonction de tâche est construite à partir de la différence entre les informations visuelles courantes et désirées. Cette approche est appelée asservissement visuel 2D car le contrôle s'effectue dans l'image [Cha90],[BE92] et [NPK83]. Cette méthode a été aussi appliquée à une caméra déportée.

Définition III.2 : Notion de ρ -admissibilité

Une fonction de tâche $\mathbf{e}(\mathbf{q}, t)$ de classe C^k , $k = 1$, d'un ouvert de $\mathbf{R}^n \times \mathbf{R}$ dans \mathbf{R}^n est ρ -admissible sur l'ensemble $C_{\rho,T}$ et sur l'intervalle de temps $[0, T]$ si

et seulement si la fonction $\mathbf{F}(\mathbf{q}, t) = \mathbf{e}(\mathbf{q}, t)$ admet un difféomorphisme de classe C^k de $C_{\rho,T}$ sur la boule fermée $B\rho \times [0, T]$, centrée en 0 et de rayon ρ .

Définition III.3 équivalente à la notion de ρ -admissibilité

Soit $C_{\rho,T}$ l'ensemble des couples (q, t) et $B\rho$ une boule fermée de centre O et de rayon ρ . Si $T < +\infty$, la propriété de ρ -admissibilité est équivalente aux trois propositions suivantes :

1. $C_{\rho,T}$ est une composante connexe par arc non vide de $F^{-1}(B\rho \times [0, T])$,
2. $C_{\rho,T}$ est une partie fermée de $\mathbf{R}^n \times \mathbf{R}$,
3. Pour tout couple (\mathbf{q}, t) appartenant à $C_{\rho,T}$:
 - $\det \left(\frac{\partial \mathbf{e}}{\partial \mathbf{q}}(q, t) \right) \neq 0$,
 - $\left\| \left(\frac{\partial \mathbf{e}}{\partial \mathbf{q}} \right)^{-1} \right\| < m_{\rho,T} < \infty$,
 - $\left\| \frac{\partial \mathbf{e}}{\partial \mathbf{q}} \right\| < m'_{\rho,T} < \infty$,

Si $T = +\infty$, ces conditions sont seulement suffisantes.

Bien que cette définition permette de définir des conditions suffisantes pour l'existence d'un lien régulier entre l'espace de la tâche et l'espace d'état, elle n'est pas facilement utilisable en pratique. Le théorème suivant, proposé par [CS91], est d'un intérêt pratique beaucoup plus grand. Il permet de garantir la ρ -admissibilité au voisinage d'une trajectoire de référence, à partir de la condition essentielle de régularité du jacobien de la tâche.

Théorème III.2 : Condition de ρ -admissibilité

Soit $\{\mathbf{e}(\mathbf{q}(t), t); \mathbf{q}_0\}$ une fonction de tâche admettant une trajectoire idéale $\mathbf{q}_r(t)$ définie sur l'intervalle de temps $[0, T]$. Si

- $\mathbf{e}(\mathbf{q}(t), t)$ est de classe C^2 ;
 - la matrice jacobienne de la tâche $\dot{\mathbf{e}} = \frac{\partial \mathbf{e}}{\partial \mathbf{q}}$ est inversible au voisinage de $\mathbf{q}_r(t)$;
 - l'horizon de temps reste fini $T < +\infty$;
- il existe un rayon $\rho > 0$ tel que la tâche considérée soit ρ -admissible.

La condition ρ -admissibilité permet de prouver l'existence d'une application bijective c'est-à-dire l'existence d'un difféomorphisme local \mathbf{F} . Ce difféomorphisme permet alors d'établir un lien régulier entre l'espace d'état et l'espace capteur dans lequel s'exprime la commande. Comme nous le rappellerons dans la section suivante, la plupart des lois de commande référencée – vision, visant à faire exécuter au robot une trajectoire de référence $\mathbf{q}_r(t)$, consistent à réguler à zéro la fonction de tâche, à l'aide des techniques souvent très complexes. Bien que la simplicité de telles approches permette dans de nombreux cas d'apporter

une solution satisfaisante au problème, elle est toutefois limitée dans son champ d'application. la notion de ρ -admissibilité permet de montrer d'une part qu'il y a équivalence entre la synthèse de la loi de commande dans l'espace d'état et la synthèse dans l'espace capteur, et d'autre part que la trajectoire de référence $\mathbf{q}_r(t)$ est unique. Finalement, la condition essentielle permettant d'établir la condition de ρ -admissibilité est de définir $\mathbf{e}(\mathbf{q}(t), t)$ de telle sorte que sa jacobienne soit inversible au voisinage de la trajectoire solution.

3.3.2 Convergence exponentielle de la fonction de tâche

La régulation à zéro de la fonction de tâche est réalisée par une commande cinématique garantissant une décroissance exponentielle de \mathbf{e} . Cette convergence vers zéro garantit l'exécution du mouvement de la caméra de la position initiale vers la position désirée. La décroissance exponentielle est obtenue pour chaque composante de \mathbf{e} en vérifiant l'équation différentielle du premier ordre suivante :

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \quad (3.4)$$

où $\lambda > 0$ est un gain scalaire positif et correspond directement à l'inverse de la constante de temps de la décroissance exponentielle de chaque composante de \mathbf{e} .

3.3.2.1 Synthèse de la commande.

La dérivée par rapport au temps de la fonction de tâche dans le cas où la matrice de combinaison \mathbf{C} est constante s'écrit :

$$\dot{\mathbf{e}} = \mathbf{C}\dot{\mathbf{s}} \quad (3.5)$$

En utilisant la relation (2.2) que nous rappelons $\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{T}_{RC/R}$, il vient alors que :

$$\dot{\mathbf{e}} = \mathbf{C}\mathbf{L}_s \mathbf{T}_{RC/R} \quad (3.6)$$

Ou bien dans le cas plus général, en tenant compte de l'équation (3.1) cette expression peut s'écrire directement en fonction du torseur cinématique $\mathbf{T}_{RC/R}$ du capteur visuel selon la relation :

$$\dot{\mathbf{e}}(\mathbf{r}(t), t) = \mathbf{C}\dot{\mathbf{s}}(\mathbf{r}(t), t) - \mathbf{C}\mathbf{L}_s \mathbf{T}_{RC/R} + \frac{\partial \mathbf{e}}{\partial t} \quad (3.7)$$

Où $\frac{\partial \mathbf{e}}{\partial t} = \mathbf{C} \frac{\partial \mathbf{s}}{\partial t}$ représente les variations de \mathbf{e} dues à la variation propre de \mathbf{s} . Il est nécessaire de disposer d'au moins six informations visuelles indépendantes pour contrôler la situation complète de la caméra. Dans ce cas, la matrice d'interaction \mathbf{L}_s sera alors de rang plein. La commande cinématique garantissant la décroissance exponentielle découplée de la fonction de tâche \mathbf{e} est :

$$\mathbf{T}_{RC/R} = -(\mathbf{CL}_s)^{-1} \left(\lambda \mathbf{e} + \frac{\partial \mathbf{e}}{\partial t} \right) \quad (3.8)$$

La matrice d'interaction dépend généralement de paramètres mal connus tels que l'information de la profondeur de la cible par rapport au repère de la caméra. La mise en oeuvre d'une telle commande est purement théorique, car elle suppose que :

- les dynamiques du système soient négligeables et n'interviennent pas dans la commande,
- la matrice d'interaction \mathbf{L}_s soit parfaitement connue, ce qui implique la connaissance des paramètres du capteur de vision, d'un modèle exact de son interaction avec la scène,
- le terme $\frac{\partial \mathbf{e}}{\partial t}$ représentant le mouvement propre de la cible soit connu, ce qui n'est généralement pas le cas,
- la matrice \mathbf{CL}_s soit inversible, carrée, et être telle que \mathbf{e} soit de dimension au moins égale au nombre de degrés de liberté n du robot.

Pour la pratique, la matrice d'interaction \mathbf{L}_s est construite sur des fortes hypothèses qui l'éloignent de la réalité car elle dépend des paramètres intrinsèques de la caméra qui sont dans la plupart des cas estimés. Pour cette raison, l'expression (3.8) est usuellement réécrite en faisant apparaître des valeurs estimées de ces grandeurs de la manière suivante :

$$\mathbf{T}_{RC/R} = -\left(\mathbf{C} \widehat{\mathbf{L}}_s\right)^{-1} \left(\lambda \mathbf{e} + \frac{\widehat{\partial \mathbf{e}}}{\partial t} \right) \quad (3.9)$$

Analyse de stabilité

La méthode de Lyapunov peut être utilisée pour étudier la stabilité du système en boucle fermée. La fonction candidate de Lyapunov choisie, est la norme au carré de la fonction de tâche définie par $\Gamma = \frac{1}{2} \|\mathbf{e}(t)\|^2$. L'analyse consiste alors à démontrer les conditions permettant de garantir la décroissance de Γ en fonction du temps. En dérivant Γ , il en résulte l'expression $\dot{\Gamma} = \mathbf{e}^T \dot{\mathbf{e}}$; où la variation $\dot{\mathbf{e}}$ de la fonction de tâche du système bouclé est obtenue en utilisant (3.9) dans (3.7) on obtient :

$$\dot{\mathbf{e}} = -\lambda \mathbf{CL}_s \left(\mathbf{C} \widehat{\mathbf{L}}_s\right)^{-1} \mathbf{e} - \mathbf{CL}_s \left(\mathbf{C} \widehat{\mathbf{L}}_s\right)^{-1} \frac{\widehat{\partial \mathbf{e}}}{\partial t} + \frac{\partial \mathbf{e}}{\partial t} \quad (3.10)$$

Dans le cas d'une cible fixe (ce qui correspond à une tâche de positionnement), $\left(\frac{\partial \mathbf{e}}{\partial t} = \frac{\widehat{\partial \mathbf{e}}}{\partial t} = 0\right)$ nous obtenons $\dot{\Gamma} = -\lambda \mathbf{e}^T \mathbf{CL}_s \left(\mathbf{C} \widehat{\mathbf{L}}_s\right)^{-1} \mathbf{e}$; la condition de positivité est alors :

$$\mathbf{C}\mathbf{L}_s \left(\mathbf{C} \widehat{\mathbf{L}}_s \right)^{-1} > 0 \quad (3.11)$$

Cette condition de stabilité apparaît directement comme une condition suffisante pour assurer la décroissance de l'erreur \mathbf{e} . D'autre part, la décroissance exponentielle de la fonction de tâche peut également être retrouvée sans cette hypothèse sous les conditions : $\mathbf{L}_s = \widehat{\mathbf{L}}_s$ et $\left(\frac{\partial \mathbf{e}}{\partial t} = \frac{\partial \widehat{\mathbf{e}}}{\partial t} \right)$. Différents choix ont été proposés pour les matrices \mathbf{C} et $\widehat{\mathbf{L}}_s$ dans le cas de l'asservissement visuel 2D. Pour simplifier les développements, nous ferons à nouveau l'hypothèse que la cible est fixe.

La loi de commande (3.10) semble donc être très prometteuse puisqu'elle est globalement asymptotiquement stable dans l'espace de la fonction de tâche \mathbf{e} . De plus, l'équation (3.11) assure le comportement spécifié $\dot{\mathbf{e}} = -\lambda \mathbf{e}$, à savoir une convergence exponentielle découplée de chaque composante de \mathbf{e} , à partir du moment où \mathbf{L}_{s*} et \mathbf{L}_s ne sont pas entachées d'erreur.

Cela dit, pour démontrer la stabilité dans l'espace des configurations, à savoir SE3 ou encore l'espace des informations visuelles \mathbf{s} , nous devons démontrer que $\mathbf{e} = 0$ si et seulement si $\mathbf{s} = \mathbf{s}^*$. Cette dernière condition devient fausse dès que

$$(\mathbf{s} - \mathbf{s}^*) \in \text{Ker } \mathbf{L}_{s*}^+ \quad (3.12)$$

Dans ce cas, on a alors $\mathbf{e} = 0$, alors que \mathbf{s} diffèrent de \mathbf{s}^* , ce qui correspond à un minimum local. En d'autres termes, la fonction de tâche \mathbf{e} choisie assure un difféomorphisme avec SE3, mais ce difféomorphisme n'est pas global (i.e. valide dans tout SE3) mais seulement local dès qu'il existe une configuration telle que (3.12) est vérifiée.

3.3.2.2 Synthèse de commandes dans les différents espaces

Cas ou $k=6$

Le cas d'un vecteur \mathbf{s} de dimension 6 est obtenu pour une cible constituée de 3 points non alignés. Dans ce cas, on peut faire le choix simple $\mathbf{C} = \mathbf{I}_6$. Le comportement du vecteur \mathbf{s} est similaire à celui de \mathbf{e} , toutes les composantes de \mathbf{s} ont une décroissance exponentielle découplée. La loi de commande devient alors :

$$\mathbf{T}_{RC/R} = -\lambda \widehat{\mathbf{L}}_s^{-1} \mathbf{e} = -\lambda \widehat{\mathbf{L}}_s^{-1} (\mathbf{s} - \mathbf{s}^*) \quad (3.13)$$

avec pour condition de stabilité

$$\mathbf{L}_s \widehat{\mathbf{L}}_s^{-1} > 0 \quad (3.14)$$

L'estimation de la valeur de \mathbf{L}_s à chaque itération de la loi de commande permet de rapprocher la dynamique de l'erreur à un comportement exponentiel idéal :

$$\dot{\mathbf{s}} = -\lambda \mathbf{L}_s \widehat{\mathbf{L}}_s^{-1} (\mathbf{s} - \mathbf{s}^*) \quad (3.15)$$

La trajectoire en boucle fermée du système est alors donnée par :

$$\mathbf{s} = \mathbf{s}^* + (\mathbf{s}_0 - \mathbf{s}^*) \mathbf{e}^{-\lambda \mathbf{L}_s \widehat{\mathbf{L}}_s^{-1} t} \approx \mathbf{s}^* + (\mathbf{s}_0 - \mathbf{s}^*) \mathbf{e}^{-\lambda t} \quad (3.16)$$

Où \mathbf{s}_0 est la valeur initiale des coordonnées des points dans l'image. Cette équation représente l'équation paramétrique d'une droite dans l'espace de sortie de la caméra. Outre le problème de mesure et d'estimation de la matrice d'interaction, deux difficultés limitent l'application de cette méthode :

1. Certaines configurations existent où la matrice d'interaction \mathbf{L}_s est singulière [Mic93] et [Pap95].
2. L'erreur peut être nulle ($\mathbf{e} = \mathbf{s} - \mathbf{s}^*$) sans que la situation désirée n'ait été atteinte. Une même image contenant la projection de trois points de la cible peut en effet être obtenue pour plusieurs situations différentes de la caméra [MD89].

Cas où $k > 6$

Afin d'éviter les problèmes décrits dans le paragraphe précédent, on peut utiliser la redondance des informations visuelles 2D, en considérant au moins 4 points non alignés ($k > 6$). On recherche une matrice de combinaison \mathbf{C} constante, de dimensions $6 \times k$ et de rang plein égal à 6. Une première approche consiste à choisir pour \mathbf{C} la pseudo-inverse de Moore- Penrose d'une approximation de la matrice d'interaction à la position désirée [BE92] :

$$\mathbf{C} = \widehat{\mathbf{L}}_{s/s^*}^+ \quad (3.17)$$

Avec cette nouvelle expression de la matrice de combinaison \mathbf{C} , la loi de commande (3.13) s'écrit alors :

$$\mathbf{T}_{RC/R} = -\lambda \left(\widehat{\mathbf{L}}_{s/s^*}^+ \widehat{\mathbf{L}}_s \right)^{-1} \mathbf{e} \quad (3.18)$$

et l'équation en boucle fermée du système devient :

$$\dot{\mathbf{s}} = -\lambda \mathbf{L}_s \left(\widehat{\mathbf{L}}_{s/s^*}^+ \widehat{\mathbf{L}}_s \right)^{-1} \widehat{\mathbf{L}}_{s/s^*}^+ (\mathbf{s} - \mathbf{s}^*) \quad (3.19)$$

Si nous devons choisir $\widehat{\mathbf{L}}_s$ comme la pseudo – inverse à droite de \mathbf{C} , on obtient :

$$\dot{\mathbf{s}} = -\lambda \mathbf{L}_s \widehat{\mathbf{L}}_{s/s^*}^+ (\mathbf{s} - \mathbf{s}^*) \quad (3.20)$$

L'annulation de $\mathbf{e} = \widehat{\mathbf{L}}_{s/s^*}^+ (\mathbf{s} - \mathbf{s}^*)$ dans le cas où $(\mathbf{s} - \mathbf{s}^*) \in \text{Ker } \mathbf{L}_{s^*}^+$ ne permet pas de garantir la convergence de \mathbf{s} vers \mathbf{s}^* . En pratique, on retrouve ces cas de figure que si des forts mouvements en rotation sont imposés. Afin d'éviter ces problèmes, deux alternatives ont été proposées :

1. La première alternative consiste à définir la matrice \mathbf{C} comme la matrice identité \mathbf{I}_k d'ordre k . Dans ce cas de figure, pour la définition de la loi de commande, on utilise une expression similaire à la relation (3.18) mais en remplaçant le calcul de l'inverse de la matrice d'interaction par sa pseudo-inverse [KH96] : $\widehat{\mathbf{L}}_s^+ = \left(\widehat{\mathbf{L}}_s \widehat{\mathbf{L}}_s \right)^{-1} \widehat{\mathbf{L}}_s$
2. La deuxième consiste tout simplement à définir directement \mathbf{C} comme la pseudo- inverse de la matrice d'interaction à l'instant courant : $\mathbf{C} = \widehat{\mathbf{L}}_s^+$.

Dans les deux cas, le torseur cinématique de la caméra vérifie la relation :

$$\mathbf{T}_{RC/R} = -\lambda \widehat{\mathbf{L}}_s^+ (\mathbf{s} - \mathbf{s}^*) \quad (3.21)$$

La matrice \mathbf{C} n'étant plus constante, la condition de stabilité (3.14) ne peut plus s'appliquer directement pour la seconde méthode. De toute manière si le comportement de \mathbf{s} est considéré, les deux méthodes conduisent à la même dynamique du système en boucle fermée :

$$\dot{\mathbf{s}} = -\lambda \mathbf{L}_s \widehat{\mathbf{L}}_s^+ (\mathbf{s} - \mathbf{s}^*) \quad (3.22)$$

La fonction de tâche ainsi définie ne vérifie plus la condition de ρ - admissibilité introduite dans les paragraphes précédents [CS91]. La convergence de \mathbf{s} vers \mathbf{s}^* ne permet plus de garantir la stabilité du capteur de vision à la situation désirée.

On ne saurait terminer cette partie sans parler de l'importance du conditionnement de la matrice d'interaction. Un bon conditionnement de cet matrice garantit une bonne précision du lien entre les informations visuelles et le mouvement de la caméra.

3.4 Discussion

Pour effectuer une tâche à six ddl, il faut utiliser au moins trois points et empiler les matrices associées à chaque point. L'étude exposée dans [Mic93] montre que certaines configurations entraînent une perte de rang de \mathbf{L}_s . Ces singularités

sont telles que les trois points soient alignés dans l'image ou que le centre optique \mathbf{C} de la caméra appartienne à la surface du cylindre défini par le cercle circonscrit aux trois points (voir figure 3.1). Il est donc difficile d'assurer que, quelle que soit la position initiale choisie, le mouvement du robot évitera de traverser une singularité isolée. Les auteurs concluent qu'il faut un minimum de quatre points pour éviter certaines singularités liées à la configuration de la caméra par rapport aux primitives ponctuelles considérées.

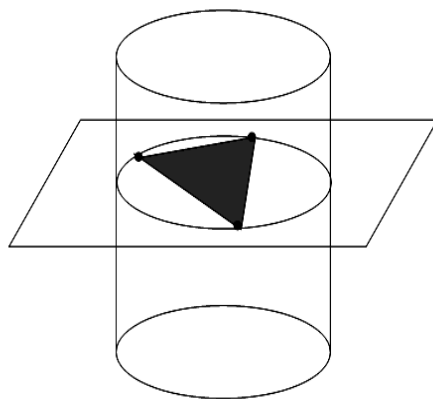


FIGURE 3.1 – cylindre de singularité.

De plus, il existe généralement quatre situations distinctes entre la caméra et la scène telles que l'image de trois points soit la même [MD89]. La minimisation de $\|\mathbf{s} - \mathbf{s}^*\|$ peut donc amener le robot à l'un des quatre minima globaux tels que $\|\mathbf{s} - \mathbf{s}^*\| = 0$. Ainsi, dans ce cas, on peut très bien avoir $\mathbf{s} - \mathbf{s}^* = 0$ sans que la situation atteinte soit celle qui a été spécifiée.

D'autres problèmes liés au choix des primitives visuelles (et donc à la définition de la matrice d'interaction) ont été mis en évidence dans la littérature, en particulier lorsque l'objet observé est un objet représenté par un ensemble de points :

- Convergence vers un minimum local [Cha98],
- Sensibilité différente de la boucle de contrôle suivant le nombre de points considérés et leurs profondeurs par rapport à la caméra [Has98],
- Choix de la matrice d'interaction : l'utilisation d'une matrice d'interaction estimée à chaque itération nécessite de déterminer la profondeur des points considérés ce qui pose des problèmes rencontrés en asservissement visuel 3D. Le choix souvent retenu du jacobien image calculé sur la position désirée peut entraîner des problèmes de convergence et de stabilité si l'erreur initiale est importante (stabilité et convergence assurées seulement dans un voisinage de la position désirée).

Afin d'éviter les situations où la matrice d'interaction dégénère, le choix d'autres primitives visuelles est proposé dans [Cha98]. En utilisant des primitives de type droite, l'auteur montre que les singularités de la matrice d'interaction sont évitées. Ceci est dû en particulier à un partitionnement de l'asservissement visuel séparant les rotations des translations. Ce partitionnement est également obtenu avec l'asservissement hybride présenté dans [Deg98] et [MCB99]. L'approche proposée par Corke et al [CH01] fournit aussi une solution au problème de singularités de la matrice d'interaction.

Nous avons proposé au chapitre précédent, un choix d'informations visuelles basées sur les moments dont le centre de gravité a été décalé. De ce choix, il en résulte une matrice d'interaction que nous voulons non singulière. Si la matrice $\mathbf{L}_{rij}^{\parallel}$ correspondant au mouvement de rotation le long de l'axe x et l'axe y est non singulière :

$$\mathbf{L}_{rij}^{\parallel} = \begin{bmatrix} r_{iwxsh1/P1} & r_{iwysh1/P1} \\ r_{jwxsh1/P2} & r_{jwysh1/P2} \end{bmatrix} \quad (3.23)$$

Alors la matrice d'interaction relative aux informations visuelles $\mathbf{s} = (x_n \ y_n \ a_n \ r_i \ r_j \ \theta)$ sera non singulière.

3.5 Matrice d'interaction relative au point P_1

Sachant que le point \mathbf{P}_1 peut être choisi dans la direction du vecteur \mathbf{n} et avec pour coordonnées x_{sh1} et y_{sh1} que nous rappelons ici (voir figure 2.3) :

$$x_{sh1} = x_g + a \cos \theta$$

$$y_{sh1} = y_g + a \sin \theta$$

Alors les matrices d'interactions correspondantes à x_{sh1} , y_{sh1} et " a " peuvent s'écrire :

$$\begin{cases} \mathbf{L}_{x_{sh1}} = \mathbf{L}_{x_g} + \mathbf{L}_a \cos \theta - a \mathbf{L}_\theta \sin \theta \\ \mathbf{L}_{y_{sh1}} = \mathbf{L}_{y_g} + \mathbf{L}_a \sin \theta + a \mathbf{L}_\theta \cos \theta \\ \mathbf{L}_a = \frac{1}{4} (\mu_{20} + \mu_{02})^{-\frac{3}{4}} (\mathbf{L}_{\mu_{20}} + \mathbf{L}_{\mu_{02}}) \end{cases}$$

Appliquons les résultats de l'équation (2.49) sur les moments d'ordre 2, qui sont μ_{20} , μ_{02} et μ_{11} (rappelons que les moments d'ordre 1 μ_{10} et μ_{01} sont nuls), on obtient alors :

$$\mathbf{L}_{\mu_{20}} = (\mu_{20_{vx}} \quad -B\mu_{20} \quad u_{20} \quad t_{20} \quad s_{20} \quad 2\mu_{11})$$

$$\mathbf{L}_{\mu_{02}} = (-A\mu_{02} \quad \mu_{02_{vy}} \quad u_{02} \quad t_{02} \quad s_{02} \quad -2\mu_{11})$$

$$\mathbf{L}_{\mu_{11}} = (\mu_{11vx} \quad \mu_{11vy} \quad u_{11} \quad t_{11} \quad s_{11} \quad \mu_{02} - \mu_{20})$$

où

$$\begin{cases} \mu_{20vx} = -3A\mu_{20} - 2B\mu_{11} \\ \mu_{02vy} = -2A\mu_{11} - 3B\mu_{02} \\ \mu_{11vx} = -2A\mu_{11} - B\mu_{02} \\ \mu_{11vy} = -A\mu_{20} - 2B\mu_{02} \\ u_{ij} = -As_{ij} + Bt_{ij} + 4C\mu_{ij} \end{cases} \quad \text{et} \quad \begin{cases} s_{20} = -7x_g\mu_{20} - 5\mu_{30} \\ s_{02} = -5(x_g\mu_{02} + \mu_{12}) - 2y_g\mu_{11} \\ s_{11} = -6x_g\mu_{11} - 5\mu_{21} - y_g\mu_{20} \\ t_{20} = -5(y_g\mu_{20} + \mu_{21}) - 2x_g\mu_{11} \\ t_{11} = 6y_g\mu_{11} + 5\mu_{12} + x_g\mu_{02} \end{cases}$$

Nous rappelons aussi la matrice d'interaction relatives a \mathbf{L}_θ

$$\mathbf{L}_\theta = (\alpha_{vx} \quad \alpha_{vy} \quad \alpha_{vz} \quad \alpha_{wx} \quad \alpha_{wy} \quad -1)$$

où

$$\begin{cases} \alpha_{vx} = -aA + bB \\ \alpha_{vy} = -cA - aB \\ \alpha_{vz} = -A\alpha_{wy} + B\alpha_{wx} \\ \alpha_{wx} = -bx_g + ay_g + d \\ \alpha_{wy} = ax_g - cy_g + e \end{cases}$$

et

$$\begin{cases} \Delta = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \\ a = \mu_{11}(\mu_{20} + \mu_{02})/\Delta \\ b = (2\mu_{11}^2 + \mu_{02}(\mu_{02} - \mu_{20}))/\Delta \\ c = (2\mu_{11}^2 + \mu_{20}(\mu_{20} - \mu_{02}))/\Delta \\ d = 5(\mu_{12}(\mu_{20} - \mu_{02}) + \mu_{11}(\mu_{03} - \mu_{21}))/\Delta \\ e = 5(\mu_{21}(\mu_{02} - \mu_{20}) + \mu_{11}(\mu_{30} - \mu_{12}))/\Delta \end{cases}$$

D'une façon similaire, en considérant les moments d'ordre 1, m_{10} et m_{01} , on peut obtenir facilement les matrices d'interaction associées aux coordonnées du centre de gravité de l'objet dans l'image $x_g = (m_{10}/m_{00})$ et $y_g = (m_{01}/m_{00})$. On obtient :

$$\mathbf{L}_{x_g} = \begin{pmatrix} -1/z_g & 0 & x_{gvz} & x_{gwx} & x_{gwy} & y_g \end{pmatrix} \quad (3.24)$$

et

$$\mathbf{L}_{y_g} = \begin{pmatrix} 0 & -1/z_g & y_{gvz} & y_{gwx} & y_{gwy} & -x_g \end{pmatrix} \quad (3.25)$$

où

$$\begin{cases} x_{gvz} = x_g/z_g + 4(A\eta_{20} + B\eta_{11}) \\ y_{gvz} = y_g/z_g + 4(A\eta_{11} + B\eta_{02}) \\ x_{gwx} = -y_{gwy} = x_g y_g + 4\eta_{11} \\ x_{gwy} = -(1 + x_g^2 + 4\eta_{20}) \\ y_{gwx} = 1 + y_g^2 + 4\eta_{02} \end{cases}$$

η_{20} , η_{02} , et η_{11} sont les moments centrés normalisés d'ordre 2 :

$$\eta_{ij} = \mu_{ij}/m_{00} \text{ avec } \begin{cases} \mu_{20} = m_{20} - m_{00}x_g^2 \\ \mu_{02} = m_{02} - m_{00}y_g^2 \\ \mu_{11} = m_{11} - m_{00}x_g y_g \end{cases}$$

Comme

$$\mathbf{L}_{x_{sh1}} = \mathbf{L}_{x_g} + \mathbf{L}_a \cos \theta - a\mathbf{L}_\theta \sin \theta \quad (3.26)$$

on en déduit

$$\mathbf{L}_{x_{sh1}} = (x_{sh1/vx} \ x_{sh1/vy} \ x_{sh1/vz} \ x_{sh1/wx} \ x_{sh1/wy} \ x_{sh1/wz}) \quad (3.27)$$

avec

$$\begin{cases} x_{sh1/vx} = -\frac{1}{z_g} + \frac{1}{4}a^3(\mu_{20vx} - A\mu_{02}) \cos \theta - a\alpha_{vx} \sin \theta \\ x_{sh1/vy} = \frac{1}{4}a^3(\mu_{02vy} - B\mu_{20}) \cos \theta - a\alpha_{vy} \sin \theta \\ x_{sh1/vz} = x_{gvz} + \frac{1}{4}a^3(\mu_{20} + \mu_{02}) \cos \theta - a\alpha_{vz} \sin \theta \\ x_{sh1/wx} = x_{gwx} + \frac{1}{4}a^3(t_{20} + t_{02}) \cos \theta - a\alpha_{wx} \sin \theta \\ x_{sh1/wy} = x_{gwy} + \frac{1}{4}a^3(s_{20} + s_{02}) \cos \theta - a\alpha_{wy} \sin \theta \\ x_{sh1/wz} = y_g + a \sin \theta \end{cases}$$

et comme

$$\mathbf{L}_{y_{sh1}} = \mathbf{L}_{y_g} + \mathbf{L}_a \sin \theta + a\mathbf{L}_\theta \cos \theta \quad (3.28)$$

on en déduit

$$\mathbf{L}_{y_{sh1}} = (y_{sh1/vx} \ y_{sh1/vy} \ y_{sh1/vz} \ y_{sh1/wx} \ y_{sh1/wy} \ y_{sh1/wz}) \quad (3.29)$$

avec

$$\begin{cases} y_{sh1/vx} = \frac{1}{4}a^3(\mu_{20vx} - A\mu_{02}) \sin \theta + a\alpha_{vx} \cos \theta \\ y_{sh1/vy} = -\frac{1}{z_g} + \frac{1}{4}a^3(\mu_{02vy} - B\mu_{20}) \sin \theta + a\alpha_{vy} \cos \theta \\ y_{sh1/vz} = y_{gvz} + \frac{1}{4}a^3(\mu_{20} + \mu_{02}) \sin \theta + a\alpha_{vz} \cos \theta \\ y_{sh1/wx} = y_{gwx} + \frac{1}{4}a^3(t_{20} + t_{02}) \sin \theta + a\alpha_{wx} \cos \theta \\ y_{sh1/wy} = y_{gwy} + \frac{1}{4}a^3(s_{20} + s_{02}) \sin \theta + a\alpha_{wy} \cos \theta \\ y_{sh1/wz} = -x_g - a \cos \theta \end{cases}$$

La variation $\dot{\mu}_{pq}$ des moments centrés se déduit de celle des moments classiques par la relation ci-après (voir chapitre II) :

$$\dot{\mu}_{pq} = \iint_{\Omega_{int}} \left[\frac{\partial k}{\partial x} (\dot{x} - \dot{x}_{sh1}) + \frac{\partial k}{\partial y} (\dot{y} - \dot{y}_{sh1}) + k(x, y) \left(\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} \right) \right] dx dy \quad (3.30)$$

Sachant que :

$$\dot{\mu}_{pq} = \mathbf{L}_{\mu_{pq}} \begin{bmatrix} \mathbf{v} \\ \omega \end{bmatrix} \quad (3.31)$$

On en déduit la matrice d'interaction relative au point \mathbf{P}_1 , $\mathbf{L}_{\mu_{pq}} [\mathbf{P}_1]$:

$$\mathbf{L}_{\mu_{pq}} [P1] = \begin{bmatrix} \mu_{v_x P1} & \mu_{v_y P1} & \mu_{v_z P1} & \mu_{w_x P1} & \mu_{w_y P1} & \mu_{w_z P1} \end{bmatrix} \quad (3.32)$$

avec

$$\left\{ \begin{array}{l} \mu_{v_x P1} = -(p+1) A \mu_{pq}^{sh1} - p B \mu_{p-1, q+1}^{sh1} - p x_{sh1/v_x} \mu_{p-1, q}^{sh1} - q y_{sh1/v_x} \mu_{p, q-1}^{sh1} \\ \quad - p (A x_{sh1} + B y_{sh1} + C) \mu_{p-1, q}^{sh1} \\ \mu_{v_y P1} = -q A \mu_{p+1, q-1}^{sh1} - (q+1) B \mu_{pq}^{sh1} - p x_{sh1/v_y} \mu_{p-1, q}^{sh1} - q y_{sh1/v_y} \mu_{p, q-1}^{sh1} \\ \quad - q (A x_{sh1} + B y_{sh1} + C) \mu_{p, q-1}^{sh1} \\ \mu_{v_z P1} = (p+q+2) C \mu_{pq}^{sh1} + (p+q+3) A \mu_{p+1, q}^{sh1} + (2p+q+3) A x_{sh1} \mu_{pq}^{sh1} \\ \quad + q y_{sh1} A \mu_{p+1, q-1}^{sh1} + p (A x_{sh1}^2 + B x_{sh1} y_{sh1} + C x_{sh1} - x_{sh1/v_z}) \mu_{p-1, q}^{sh1} \\ \quad + (p+q+3) B \mu_{p, q+1}^{sh1} + p x_{sh1} B \mu_{p-1, q+1}^{sh1} \\ \quad + (p+2q+3) y_{sh1} \mu_{pq}^{sh1} + q (B y_{sh1}^2 + A x_{sh1} y_{sh1} + C y_{sh1} - y_{sh1/v_z}) \mu_{p, q-1}^{sh1} \\ \mu_{w_x P1} = (p+q+3) \mu_{p, q+1}^{sh1} + p x_{sh1} \mu_{p-1, q+1}^{sh1} + (p+2q+3) y_{sh1} \mu_{pq}^{sh1} \\ \quad + p (x_{sh1} y_{sh1} - x_{sh1/w_x}) \mu_{p-1, q}^{sh1} + q (1 + y_{sh1}^2 - y_{sh1/w_x}) \mu_{p, q-1}^{sh1} \\ \mu_{w_y P1} = -(p+q+3) \mu_{p+1, q}^{sh1} - (2p+q+3) x_{sh1} \mu_{pq}^{sh1} - q y_{sh1} \mu_{p+1, q-1}^{sh1} \\ \quad - p (1 + x_{sh1}^2 + x_{sh1/w_x}) \mu_{p-1, q}^{sh1} - q (x_{sh1} y_{sh1} + y_{sh1/w_x}) \mu_{p, q-1}^{sh1} \\ \mu_{w_z P1} = p \mu_{p-1, q+1}^{sh1} - q \mu_{p+1, q-1}^{sh1} + p (y_{sh1} - x_{sh1/w_z}) \mu_{p-1, q}^{sh1} - q (x_{sh1} - y_{sh1/w_z}) \mu_{p, q-1}^{sh1} \end{array} \right.$$

Où les μ_{pq}^{sh1} se calculent suivant la relation (2.61) que nous pouvons réécrire sous la forme :

$$\begin{aligned} \mu_{pq}^{sh1} &= \iint_R \left[\sum_{i=0}^p \binom{p}{i} (x - x_g)^{p-i} x_{sh1}^i \sum_{j=0}^q \binom{q}{j} (y - y_g)^{q-j} y_{sh1}^j \right] dx dy \\ &= \sum_{i=0}^p \sum_{j=0}^q \binom{p}{i} \binom{q}{j} x_{sh1}^i y_{sh1}^j \mu_{p-i, q-j} \end{aligned} \quad (3.33)$$

3.6 Matrice d'interaction relative au point \mathbf{P}_2

Sachant que le point \mathbf{P}_2 peut être choisi dans la direction du vecteur \mathbf{v} avec pour coordonnées x_{sh2} et y_{sh2} que nous rappelons ici (voir figure 2.3) :

$$\begin{aligned}x_{sh2} &= x_g + a \cos \left(\theta + \frac{\pi}{2} \right) \\ y_{sh2} &= y_g + a \sin \left(\theta + \frac{\pi}{2} \right)\end{aligned}$$

Et comme

$$\mathbf{L}_{x_{sh2}} = \mathbf{L}_a \cos \left(\theta + \frac{\pi}{2} \right) - a \mathbf{L}_\theta \sin \left(\theta + \frac{\pi}{2} \right) \quad (3.34)$$

on en déduit

$$\mathbf{L}_{x_{sh2}} = \begin{pmatrix} x_{sh2/vx} & x_{sh2/vy} & x_{sh2/vz} & x_{sh2/wx} & x_{sh2/wy} & x_{sh2/wz} \end{pmatrix} \quad (3.35)$$

Avec

$$\begin{cases} x_{sh2/vx} = -\frac{1}{z_g} + \frac{1}{4}a^3 (\mu_{20vx} - A\mu_{02}) \cos \left(\theta + \frac{\pi}{2} \right) - a\alpha_{vx} \sin \left(\theta + \frac{\pi}{2} \right) \\ x_{sh2/vy} = \frac{1}{4}a^3 (\mu_{02vy} - B\mu_{20}) \cos \left(\theta + \frac{\pi}{2} \right) - a\alpha_{vy} \sin \left(\theta + \frac{\pi}{2} \right) \\ x_{sh2/vz} = x_{g_{vz}} + \frac{1}{4}a^3 (\mu_{20} + \mu_{02}) \cos \left(\theta + \frac{\pi}{2} \right) - a\alpha_{vz} \sin \left(\theta + \frac{\pi}{2} \right) \\ x_{sh2/wx} = x_{g_{wx}} + \frac{1}{4}a^3 (t_{20} + t_{02}) \cos \left(\theta + \frac{\pi}{2} \right) - a\alpha_{wx} \sin \left(\theta + \frac{\pi}{2} \right) \\ x_{sh2/wy} = x_{g_{wy}} + \frac{1}{4}a^3 (s_{20} + s_{02}) \cos \left(\theta + \frac{\pi}{2} \right) - a\alpha_{wy} \sin \left(\theta + \frac{\pi}{2} \right) \\ x_{sh2/wz} = y_g + a \sin \left(\theta + \frac{\pi}{2} \right) \end{cases}$$

et comme

$$\mathbf{L}_{y_{sh2}} = \mathbf{L}_a \sin \left(\theta + \frac{\pi}{2} \right) + a \mathbf{L}_\theta \cos \left(\theta + \frac{\pi}{2} \right) \quad (3.36)$$

on en déduit

$$\mathbf{L}_{y_{sh2}} = \begin{pmatrix} y_{sh2/vx} & y_{sh2/vy} & y_{sh2/vz} & y_{sh2/wx} & y_{sh2/wy} & y_{sh2/wz} \end{pmatrix} \quad (3.37)$$

avec

$$\begin{cases} y_{sh2/vx} = \frac{1}{4}a^3 (\mu_{20vx} - A\mu_{02}) \sin \left(\theta + \frac{\pi}{2} \right) + a\alpha_{vx} \cos \left(\theta + \frac{\pi}{2} \right) \\ y_{sh2/vy} = -\frac{1}{z_g} + \frac{1}{4}a^3 (\mu_{02vy} - B\mu_{20}) \sin \left(\theta + \frac{\pi}{2} \right) + a\alpha_{vy} \cos \left(\theta + \frac{\pi}{2} \right) \\ y_{sh2/vz} = y_{g_{vz}} + \frac{1}{4}a^3 (\mu_{20} + \mu_{02}) \sin \left(\theta + \frac{\pi}{2} \right) + a\alpha_{vz} \cos \left(\theta + \frac{\pi}{2} \right) \\ y_{sh2/wx} = y_{g_{wx}} + \frac{1}{4}a^3 (t_{20} + t_{02}) \sin \left(\theta + \frac{\pi}{2} \right) + a\alpha_{wx} \cos \left(\theta + \frac{\pi}{2} \right) \\ y_{sh2/wy} = y_{g_{wy}} + \frac{1}{4}a^3 (s_{20} + s_{02}) \sin \left(\theta + \frac{\pi}{2} \right) + a\alpha_{wy} \cos \left(\theta + \frac{\pi}{2} \right) \\ y_{sh2/wz} = -x_g - a \cos \left(\theta + \frac{\pi}{2} \right) \end{cases}$$

On en déduit la matrice d'interaction relative au point \mathbf{P}_2 , $\mathbf{L}_{\mu_{pq}} [\mathbf{P}_2]$:

$$\mathbf{L}_{\mu_{pq}} [P2] = \begin{bmatrix} \mu_{v_x P2} & \mu_{v_y P2} & \mu_{v_z P2} & \mu_{w_x P2} & \mu_{w_y P2} & \mu_{w_z P2} \end{bmatrix} \quad (3.38)$$

avec

$$\left\{ \begin{array}{l} \mu_{v_x P_2} = -(p+1) A \mu_{pq}^{sh2} - p B \mu_{p-1,q+1}^{sh2} - p x_{sh2/v_x} \mu_{p-1,q}^{sh2} - q y_{sh2/v_x} \mu_{p,q-1}^{sh2} \\ \quad - p (A x_{sh2} + B y_{sh2} + C) \mu_{p-1,q}^{sh2} \\ \mu_{v_y P_2} = -q A \mu_{p+1,q-1}^{sh2} - (q+1) B \mu_{pq}^{sh2} - p x_{sh2/v_y} \mu_{p-1,q}^{sh2} - q y_{sh2/v_y} \mu_{p,q-1}^{sh2} \\ \quad - q (A x_{sh2} + B y_{sh2} + C) \mu_{p,q-1}^{sh2} \\ \mu_{v_z P_2} = (p+q+2) C \mu_{pq}^{sh2} + (p+q+3) A \mu_{p+1,q}^{sh2} + (2p+q+3) A x_{sh2} \mu_{pq}^{sh2} \\ \quad + q y_{sh2} A \mu_{p+1,q-1}^{sh2} + p (A x_{sh2}^2 + B x_{sh2} y_{sh2} + C x_{sh2} - x_{sh2/v_z}) \mu_{p-1,q}^{sh2} \\ \quad + (p+q+3) B \mu_{p,q+1}^{sh2} + p x_{sh2} B \mu_{p-1,q+1}^{sh2} \\ \quad + (p+2q+3) y_{sh2} \mu_{pq}^{sh2} + q (B y_{sh2}^2 + A x_{sh2} y_{sh2} + C y_{sh2} - y_{sh2/v_z}) \mu_{p,q-1}^{sh2} \\ \mu_{w_x P_2} = (p+q+3) \mu_{p,q+1}^{sh2} + p x_{sh2} \mu_{p-1,q+1}^{sh2} + (p+2q+3) y_{sh2} \mu_{pq}^{sh2} \\ \quad + p (x_{sh2} y_{sh2} - x_{sh2/w_x}) \mu_{p-1,q}^{sh2} + q (1 + y_{sh2}^2 - y_{sh2/w_x}) \mu_{p,q-1}^{sh2} \\ \mu_{w_y P_2} = -(p+q+3) \mu_{p+1,q}^{sh2} - (2p+q+3) x_{sh2} \mu_{pq}^{sh2} - q y_{sh2} \mu_{p+1,q-1}^{sh2} \\ \quad - p (1 + x_{sh2}^2 + x_{sh2/w_x}) \mu_{p-1,q}^{sh2} - q (x_{sh2} y_{sh2} + y_{sh2/w_x}) \mu_{p,q-1}^{sh2} \\ \mu_{w_z P_2} = p \mu_{p-1,q+1}^{sh2} - q \mu_{p+1,q-1}^{sh2} + p (y_{sh2} - x_{sh2/w_z}) \mu_{p-1,q}^{sh2} - q (x_{sh2} - y_{sh2/w_z}) \mu_{p,q-1}^{sh2} \end{array} \right.$$

Où les μ_{pq}^{sh2} se calculent suivant la relation (2.61) que nous pouvons réécrire sous la forme :

$$\begin{aligned} \mu_{pq}^{sh2} &= \iint_R \left[\sum_{i=0}^p \binom{p}{i} (x - x_g)^{p-i} x_{sh2}^i \sum_{j=0}^q \binom{q}{j} (y - y_g)^{q-j} y_{sh2}^j \right] dx dy \\ &= \sum_{i=0}^p \sum_{j=0}^q \binom{p}{i} \binom{q}{j} x_{sh2}^i y_{sh2}^j \mu_{p-i,q-j} \end{aligned} \quad (3.39)$$

3.7 Matrice d'interaction relative à l'invariant r_{s3}

Considérons les polynômes suivants :

$$I_{s1} = -\mu_{30}^{sh} \mu_{12}^{sh} + \mu_{21}^{sh} \mu_{21}^{sh} - \mu_{03}^{sh} \mu_{21}^{sh} + \mu_{12}^{sh} \mu_{12}^{sh}$$

$$I_{s3} = 3\mu_{30}^{sh} \mu_{12}^{sh} + \mu_{30}^{sh} \mu_{30}^{sh} + 3\mu_{03}^{sh} \mu_{21}^{sh} + \mu_{03}^{sh} \mu_{03}^{sh}$$

Et l'invariant

$$r_{s3} = \frac{I_{s3}}{I_{s1}}$$

1. En considérant le point \mathbf{P}_1 on aura comme invariant r_i pour commander les rotations autour de w_x

$$r_i = \frac{I_{s31}}{I_{s11}} \quad (3.40)$$

Où

$$I_{s11} = -\mu_{30}^{sh1} \mu_{12}^{sh1} + \mu_{21}^{sh1} \mu_{21}^{sh1} - \mu_{03}^{sh1} \mu_{21}^{sh1} + \mu_{12}^{sh1} \mu_{12}^{sh1} \quad (3.41)$$

$$I_{s31} = 3\mu_{30}^{sh1} \mu_{12}^{sh1} + \mu_{30}^{sh1} \mu_{30}^{sh1} + 3\mu_{03}^{sh1} \mu_{21}^{sh1} + \mu_{03}^{sh1} \mu_{03}^{sh1} \quad (3.42)$$

En prenant les dérivées de ces équations on obtient :

$$\mathbf{L}_{I_{s11}} = -\left(\mu_{12}^{sh1} \mathbf{L}_{\mu_{30}^{sh1}} + \mu_{30}^{sh1} \mathbf{L}_{\mu_{12}^{sh1}}\right) + 2\mu_{21}^{sh1} \mathbf{L}_{\mu_{21}^{sh1}} - \left(\mu_{21}^{sh1} \mathbf{L}_{\mu_{03}^{sh1}} + \mu_{03}^{sh1} \mathbf{L}_{\mu_{21}^{sh1}}\right) + 2\mu_{12}^{sh1} \mathbf{L}_{\mu_{12}^{sh1}} \quad (3.43)$$

et

$$\mathbf{L}_{I_{s31}} = 3\left(\mu_{12}^{sh1} \mathbf{L}_{\mu_{30}^{sh1}} + \mu_{30}^{sh1} \mathbf{L}_{\mu_{12}^{sh1}}\right) + 2\mu_{30}^{sh1} \mathbf{L}_{\mu_{30}^{sh1}} + 3\left(\mu_{21}^{sh1} \mathbf{L}_{\mu_{03}^{sh1}} + \mu_{03}^{sh1} \mathbf{L}_{\mu_{21}^{sh1}}\right) + 2\mu_{03}^{sh1} \mathbf{L}_{\mu_{03}^{sh1}} \quad (3.44)$$

1. En considérant le point \mathbf{P}_2 on aura comme invariant r_j pour commander les rotations autour de wy

$$r_j = \frac{I_{s32}}{I_{s12}} \quad (3.45)$$

Où

$$I_{s12} = -\mu_{30}^{sh2} \mu_{12}^{sh2} + \mu_{21}^{sh2} \mu_{21}^{sh2} - \mu_{03}^{sh2} \mu_{21}^{sh2} + \mu_{12}^{sh2} \mu_{12}^{sh2} \quad (3.46)$$

$$I_{s32} = 3\mu_{30}^{sh2} \mu_{12}^{sh2} + \mu_{30}^{sh2} \mu_{30}^{sh2} + 3\mu_{03}^{sh2} \mu_{21}^{sh2} + \mu_{03}^{sh2} \mu_{03}^{sh2} \quad (3.47)$$

En prenant les dérivées de ces équations on obtient :

$$\mathbf{L}_{I_{s12}} = -\left(\mu_{12}^{sh2} \mathbf{L}_{\mu_{30}^{sh2}} + \mu_{30}^{sh2} \mathbf{L}_{\mu_{12}^{sh2}}\right) + 2\mu_{21}^{sh2} \mathbf{L}_{\mu_{21}^{sh2}} - \left(\mu_{21}^{sh2} \mathbf{L}_{\mu_{03}^{sh2}} + \mu_{03}^{sh2} \mathbf{L}_{\mu_{21}^{sh2}}\right) + 2\mu_{12}^{sh2} \mathbf{L}_{\mu_{12}^{sh2}} \quad (3.48)$$

et

$$\mathbf{L}_{I_{s32}} = 3\left(\mu_{12}^{sh2} \mathbf{L}_{\mu_{30}^{sh2}} + \mu_{30}^{sh2} \mathbf{L}_{\mu_{12}^{sh2}}\right) + 2\mu_{30}^{sh2} \mathbf{L}_{\mu_{30}^{sh2}} + 3\left(\mu_{21}^{sh2} \mathbf{L}_{\mu_{03}^{sh2}} + \mu_{03}^{sh2} \mathbf{L}_{\mu_{21}^{sh2}}\right) + 2\mu_{03}^{sh2} \mathbf{L}_{\mu_{03}^{sh2}} \quad (3.49)$$

Comme

$$\begin{aligned} \mu_{w_x P_1} &= (p+q+3) \mu_{p,q+1}^{sh1} + p x_{sh1} \mu_{p-1,q+1}^{sh1} \\ &\quad + (p+2q+3) y_{sh1} \mu_{pq}^{sh1} + p (x_{sh1} y_{sh1} - x_{sh1/w_x}) \mu_{p-1,q}^{sh1} + q (1 + y_{sh1}^2 - y_{sh1/w_x}) \mu_{p,q-1}^{sh1} \end{aligned} \quad (3.50)$$

,

$$\begin{aligned} \mu_{w_y P_1} = & -(p+q+3) \mu_{p+1,q}^{sh1} - (2p+q+3) x_{sh1} \mu_{pq}^{sh1} \\ & - q y_{sh1} \mu_{p+1,q-1}^{sh1} - p (1 + x_{sh1}^2 + x_{sh1/w_x}) \mu_{p-1,q}^{sh1} - q (x_{sh1} y_{sh1} + y_{sh1/w_x}) \mu_{p,q-1}^{sh1} \end{aligned} \quad (3.51)$$

et

$$\begin{aligned} \mu_{w_x P_2} = & (p+q+3) \mu_{p,q+1}^{sh2} + p x_{sh2} \mu_{p-1,q+1}^{sh2} \\ & + (p+2q+3) y_{sh2} \mu_{pq}^{sh2} + p (x_{sh2} y_{sh2} - x_{sh2/w_x}) \mu_{p-1,q}^{sh2} + q (1 + y_{sh2}^2 - y_{sh2/w_x}) \mu_{p,q-1}^{sh2} \end{aligned} \quad (3.52)$$

,

$$\begin{aligned} \mu_{w_y P_2} = & -(p+q+3) \mu_{p+1,q}^{sh2} - (2p+q+3) x_{sh2} \mu_{pq}^{sh2} \\ & - q y_{sh2} \mu_{p+1,q-1}^{sh2} - p (1 + x_{sh2}^2 + x_{sh2/w_x}) \mu_{p-1,q}^{sh2} - q (x_{sh2} y_{sh2} + y_{sh2/w_x}) \mu_{p,q-1}^{sh2} \end{aligned} \quad (3.53)$$

On calcul les différentes matrices d'interactions correspondant à w_x et w_y

1. Pour $p=3$ et $q=0$, on a

$$\begin{aligned} \mathbf{L}_{\mu_{30}^{sh1}/w_x} &= 6\mu_{31}^{sh1} + 3x_{sh1}\mu_{21}^{sh1} + 6y_{sh1}\mu_{30}^{sh1} + 3(x_{sh1}y_{sh1} - x_{sh1/w_x})\mu_{20}^{sh1} \\ \mathbf{L}_{\mu_{30}^{sh1}/w_y} &= -6\mu_{40}^{sh1} - 9x_{sh1}\mu_{30}^{sh1} - 3(1 + x_{sh1}^2 + x_{sh1/w_x})\mu_{20}^{sh1} \end{aligned} \quad (3.54)$$

$$\begin{aligned} \mathbf{L}_{\mu_{30}^{sh2}/w_x} &= 6\mu_{31}^{sh2} + 3x_{sh2}\mu_{21}^{sh2} + 6y_{sh2}\mu_{30}^{sh2} + 3(x_{sh2}y_{sh2} - x_{sh2/w_x})\mu_{20}^{sh2} \\ \mathbf{L}_{\mu_{30}^{sh2}/w_y} &= -6\mu_{40}^{sh2} - 9x_{sh2}\mu_{30}^{sh2} - 3(1 + x_{sh2}^2 + x_{sh2/w_x})\mu_{20}^{sh2} \end{aligned} \quad (3.55)$$

1. Pour $p=0$ et $q=3$, on a

$$\begin{aligned} \mathbf{L}_{\mu_{03}^{sh1}/w_x} &= 6\mu_{04}^{sh1} + 9y_{sh1}\mu_{03}^{sh1} + 3(1 + y_{sh1}^2 - y_{sh1/w_x})\mu_{02}^{sh1} \\ \mathbf{L}_{\mu_{03}^{sh1}/w_y} &= -6\mu_{13}^{sh1} - 6x_{sh1}\mu_{03}^{sh1} - 3y_{sh1}\mu_{12}^{sh1} - 3(x_{sh1}y_{sh1} + y_{sh1/w_x})\mu_{02}^{sh1} \end{aligned} \quad (3.56)$$

$$\begin{aligned} \mathbf{L}_{\mu_{03}^{sh2}/w_x} &= 6\mu_{04}^{sh2} + 9y_{sh2}\mu_{03}^{sh2} + 3(1 + y_{sh2}^2 - y_{sh2/w_x})\mu_{02}^{sh2} \\ \mathbf{L}_{\mu_{03}^{sh2}/w_y} &= -6\mu_{13}^{sh2} - 6x_{sh2}\mu_{03}^{sh2} - 3y_{sh2}\mu_{12}^{sh2} - 3(x_{sh2}y_{sh2} + y_{sh2/w_x})\mu_{02}^{sh2} \end{aligned} \quad (3.57)$$

2. Pour $p=2$ et $q=1$, on a

$$\begin{aligned}
\mathbf{L}_{\mu_{21}^{sh1}/wx} &= 6\mu_{22}^{sh1} + 2x_{sh1}\mu_{12}^{sh1} + 7y_{sh1}\mu_{21}^{sh1} + 2(x_{sh1}y_{sh1} - x_{sh1/wx})\mu_{11}^{sh1} \\
&\quad + (1 + y_{sh1}^2 - y_{sh1/wx})\mu_{20}^{sh1} \\
\mathbf{L}_{\mu_{21}^{sh1}/wy} &= -6\mu_{31}^{sh1} - 8x_{sh1}\mu_{21}^{sh1} - y_{sh1}\mu_{30}^{sh1} - 2(1 + x_{sh1}^2 + x_{sh1/wx})\mu_{11}^{sh1} \\
&\quad - (x_{sh1}y_{sh1} + y_{sh1/wx})\mu_{20}^{sh1}
\end{aligned} \tag{3.58}$$

$$\begin{aligned}
\mathbf{L}_{\mu_{21}^{sh2}/wx} &= 6\mu_{22}^{sh2} + 2x_{sh2}\mu_{12}^{sh2} + 7y_{sh2}\mu_{21}^{sh2} + 2(x_{sh2}y_{sh2} - x_{sh2/wx})\mu_{11}^{sh2} \\
&\quad + (1 + y_{sh2}^2 - y_{sh2/wx})\mu_{20}^{sh2} \\
\mathbf{L}_{\mu_{21}^{sh2}/wy} &= -6\mu_{31}^{sh2} - 8x_{sh2}\mu_{21}^{sh2} - y_{sh2}\mu_{30}^{sh2} \\
&\quad - 2(1 + x_{sh2}^2 + x_{sh2/wx})\mu_{11}^{sh2} - (x_{sh2}y_{sh2} + y_{sh2/wx})\mu_{20}^{sh2}
\end{aligned} \tag{3.59}$$

1. Pour $p=1$ et $q=2$, on a

$$\begin{aligned}
\mathbf{L}_{\mu_{12}^{sh1}/wx} &= 6\mu_{13}^{sh1} + x_{sh1}\mu_{03}^{sh1} + 7y_{sh1}\mu_{12}^{sh1} + (x_{sh1}y_{sh1} - x_{sh1/wx})\mu_{02}^{sh1} \\
&\quad + 2(1 + y_{sh1}^2 - y_{sh1/wx})\mu_{11}^{sh1} \\
\mathbf{L}_{\mu_{12}^{sh1}/wy} &= -6\mu_{22}^{sh1} - 7x_{sh1}\mu_{12}^{sh1} - 2y_{sh1}\mu_{21}^{sh1} - (1 + x_{sh1}^2 + x_{sh1/wx})\mu_{02}^{sh1} \\
&\quad - 2(x_{sh1}y_{sh1} + y_{sh1/wx})\mu_{11}^{sh1}
\end{aligned} \tag{3.60}$$

$$\begin{aligned}
\mathbf{L}_{\mu_{12}^{sh2}/wx} &= 6\mu_{13}^{sh2} + x_{sh2}\mu_{03}^{sh2} + 7y_{sh2}\mu_{12}^{sh2} + (x_{sh2}y_{sh2} - x_{sh2/wx})\mu_{02}^{sh2} \\
&\quad + 2(1 + y_{sh2}^2 - y_{sh2/wx})\mu_{11}^{sh2} \\
\mathbf{L}_{\mu_{12}^{sh2}/wy} &= -6\mu_{22}^{sh2} - 7x_{sh2}\mu_{12}^{sh2} - 2y_{sh2}\mu_{21}^{sh2} - (1 + x_{sh2}^2 + x_{sh2/wx})\mu_{02}^{sh2} \\
&\quad - 2(x_{sh2}y_{sh2} + y_{sh2/wx})\mu_{11}^{sh2}
\end{aligned} \tag{3.61}$$

Sachant que

$$r_i = \frac{I_{s31}}{I_{s11}} \tag{3.62}$$

et

$$r_j = \frac{I_{s32}}{I_{s12}} \tag{3.63}$$

on en déduit

$$\mathbf{L}_{ri} = \frac{1}{I_{s11}}\mathbf{L}_{I_{s31}} - \frac{I_{s31}}{I_{s11}^2}\mathbf{L}_{I_{s11}} \tag{3.64}$$

$$\mathbf{L}_{rj} = \frac{1}{I_{s12}} \mathbf{L}_{I_{s32}} - \frac{I_{s32}}{I_{s12}^2} \mathbf{L}_{I_{s12}} \quad (3.65)$$

Ainsi la matrice d'interaction relative a l'invariant r_{s3} suivant l'axe w_x et l'axe w_y aura la forme

$$\mathbf{L}_{rsi}^{\parallel} = \begin{bmatrix} 0 & 0 & 0 & r_{iwxsh1/P1} & r_{iwysh1/P1} & 0 \end{bmatrix}$$

et

$$\mathbf{L}_{rsj}^{\parallel} = \begin{bmatrix} 0 & 0 & 0 & r_{jwxsh2/P2} & r_{jwysh2/P2} & 0 \end{bmatrix}$$

avec

$$r_{iwxsh1/P1} = L_{ri/wx} = \frac{1}{I_{s11}} L_{I_{s31}/wx} - \frac{I_{s31}}{I_{s11}^2} L_{I_{s11}/wx}$$

$$r_{iwysh1/P1} = L_{ri/wy} = \frac{1}{I_{s11}} L_{I_{s31}/wy} - \frac{I_{s31}}{I_{s11}^2} L_{I_{s11}/wy}$$

$$r_{jwxsh2/P2} = L_{rj/wx} = \frac{1}{I_{s12}} L_{I_{s32}/wx} - \frac{I_{s32}}{I_{s12}^2} L_{I_{s12}/wx}$$

$$r_{jwysh2/P2} = L_{rj/wy} = \frac{1}{I_{s12}} L_{I_{s32}/wy} - \frac{I_{s32}}{I_{s12}^2} L_{I_{s12}/wy}$$

et

$$\begin{aligned} \mathbf{L}_{I_{S11}/wx} = & - \left(\mu_{12}^{sh1} \mathbf{L}_{\mu_{30}^{sh1}/wx} + \mu_{30}^{sh1} \mathbf{L}_{\mu_{12}^{sh1}/wx} \right) + 2\mu_{21}^{sh1} \mathbf{L}_{\mu_{21}^{sh1}/wx} \\ & - \left(\mu_{21}^{sh1} \mathbf{L}_{\mu_{03}^{sh1}/wx} + \mu_{03}^{sh1} \mathbf{L}_{\mu_{21}^{sh1}/wx} \right) + 2\mu_{12}^{sh1} \mathbf{L}_{\mu_{12}^{sh1}/wx} \end{aligned} \quad (3.66)$$

$$\begin{aligned} \mathbf{L}_{I_{S31}/wx} = & 3 \left(\mu_{12}^{sh1} \mathbf{L}_{\mu_{30}^{sh1}/wx} + \mu_{30}^{sh1} \mathbf{L}_{\mu_{12}^{sh1}/wx} \right) + 2\mu_{30}^{sh1} \mathbf{L}_{\mu_{30}^{sh1}/wx} \\ & + 3 \left(\mu_{21}^{sh1} \mathbf{L}_{\mu_{03}^{sh1}/wx} + \mu_{03}^{sh1} \mathbf{L}_{\mu_{21}^{sh1}/wx} \right) + 2\mu_{03}^{sh1} \mathbf{L}_{\mu_{03}^{sh1}/wx} \end{aligned} \quad (3.67)$$

$$\begin{aligned} \mathbf{L}_{I_{S11}/wy} = & - \left(\mu_{12}^{sh1} \mathbf{L}_{\mu_{30}^{sh1}/wy} + \mu_{30}^{sh1} \mathbf{L}_{\mu_{12}^{sh1}/wy} \right) + 2\mu_{21}^{sh1} \mathbf{L}_{\mu_{21}^{sh1}/wy} \\ & - \left(\mu_{21}^{sh1} \mathbf{L}_{\mu_{03}^{sh1}/wy} + \mu_{03}^{sh1} \mathbf{L}_{\mu_{21}^{sh1}/wy} \right) + 2\mu_{12}^{sh1} \mathbf{L}_{\mu_{12}^{sh1}/wy} \end{aligned} \quad (3.68)$$

$$\begin{aligned} \mathbf{L}_{I_{S31}/wy} = & 3 \left(\mu_{12}^{sh1} \mathbf{L}_{\mu_{30}^{sh1}/wy} + \mu_{30}^{sh1} \mathbf{L}_{\mu_{12}^{sh1}/wy} \right) + 2\mu_{30}^{sh1} \mathbf{L}_{\mu_{30}^{sh1}/wy} \\ & + 3 \left(\mu_{21}^{sh1} \mathbf{L}_{\mu_{03}^{sh1}/wy} + \mu_{03}^{sh1} \mathbf{L}_{\mu_{21}^{sh1}/wy} \right) + 2\mu_{03}^{sh1} \mathbf{L}_{\mu_{03}^{sh1}/wy} \end{aligned} \quad (3.69)$$

$$\begin{aligned} \mathbf{L}_{I_{S12}/wx} = & - \left(\mu_{12}^{sh2} \mathbf{L}_{\mu_{30}^{sh2}/wx} + \mu_{30}^{sh2} \mathbf{L}_{\mu_{12}^{sh2}/wx} \right) + 2\mu_{21}^{sh2} \mathbf{L}_{\mu_{21}^{sh2}/wx} \\ & - \left(\mu_{21}^{sh2} \mathbf{L}_{\mu_{03}^{sh2}/wx} + \mu_{03}^{sh2} \mathbf{L}_{\mu_{21}^{sh2}/wx} \right) + 2\mu_{12}^{sh2} \mathbf{L}_{\mu_{12}^{sh2}/wx} \end{aligned} \quad (3.70)$$

$$\begin{aligned} \mathbf{L}_{I_{S32}/wx} = & 3 \left(\mu_{12}^{sh2} \mathbf{L}_{\mu_{30}^{sh2}/wx} + \mu_{30}^{sh2} \mathbf{L}_{\mu_{12}^{sh2}/wx} \right) + 2\mu_{30}^{sh2} \mathbf{L}_{\mu_{30}^{sh2}/wx} \\ & + 3 \left(\mu_{21}^{sh2} \mathbf{L}_{\mu_{03}^{sh2}/wx} + \mu_{03}^{sh2} \mathbf{L}_{\mu_{21}^{sh2}/wx} \right) + 2\mu_{03}^{sh2} \mathbf{L}_{\mu_{03}^{sh2}/wx} \end{aligned} \quad (3.71)$$

$$\begin{aligned} \mathbf{L}_{I_{S12}/wy} = & - \left(\mu_{12}^{sh2} \mathbf{L}_{\mu_{30}^{sh2}/wy} + \mu_{30}^{sh2} \mathbf{L}_{\mu_{12}^{sh2}/wy} \right) + 2\mu_{21}^{sh2} \mathbf{L}_{\mu_{21}^{sh2}/wy} \\ & - \left(\mu_{21}^{sh2} \mathbf{L}_{\mu_{03}^{sh2}/wy} + \mu_{03}^{sh2} \mathbf{L}_{\mu_{21}^{sh2}/wy} \right) + 2\mu_{12}^{sh2} \mathbf{L}_{\mu_{12}^{sh2}/wy} \end{aligned} \quad (3.72)$$

$$\begin{aligned} \mathbf{L}_{I_{S32}/wy} = & 3 \left(\mu_{12}^{sh2} \mathbf{L}_{\mu_{30}^{sh2}/wy} + \mu_{30}^{sh2} \mathbf{L}_{\mu_{12}^{sh2}/wy} \right) + 2\mu_{30}^{sh2} \mathbf{L}_{\mu_{30}^{sh2}/wy} \\ & + 3 \left(\mu_{21}^{sh2} \mathbf{L}_{\mu_{03}^{sh2}/wy} + \mu_{03}^{sh2} \mathbf{L}_{\mu_{21}^{sh2}/wy} \right) + 2\mu_{03}^{sh2} \mathbf{L}_{\mu_{03}^{sh2}/wy} \end{aligned} \quad (3.73)$$

3.8 Stabilité

Nous rappelons que :

Si la matrice $\mathbf{L}_{rij}^{\parallel}$ correspondant au mouvement de rotation le long de l'axe x et l'axe y est non singulière :

$$\mathbf{L}_{rij}^{\parallel} = \begin{bmatrix} r_{iwxsh1/P1} & r_{iwysh1/P1} \\ r_{jwxsh1/P2} & r_{jwysh1/P2} \end{bmatrix} \quad (3.74)$$

Alors la matrice d'interaction relative aux informations visuelles

$\mathbf{s} = (x_n \ y_n \ a_n \ r_i \ r_j \ \theta)$ sera non singulière.

Pour que la matrice $\mathbf{L}_{rij}^{\parallel}$ soit non singulière, une condition nécessaire et suffisante est que

$$\det(\mathbf{L}_{rij}^{\parallel}) = \mathbf{L}_{ri/wx} \mathbf{L}_{rj/wy} - \mathbf{L}_{ri/wy} \mathbf{L}_{rj/wx} \neq 0 \quad (3.75)$$

3.9 Conclusion

Dans ce chapitre, nous avons étudié la commande référencée capteur des robots manipulateurs en utilisant les mesures issues de capteurs embarqués. La problématique sous-jacente, liée au caractère très non linéaire des équations cinématiques des robots, est celle de la robustesse des propriétés de stabilité associées aux commandes considérées, vis-à-vis d'incertitudes sur le modèle du capteur.

D'un point de vue théorique, nous avons étudié la stabilité des systèmes non linéaires et les résultats obtenus ont été insérés dans une boucle de commande des bras manipulateurs afin d'obtenir la loi de convergence et la condition de stabilité. Tous les résultats de simulations que nous avons obtenus dans le dernier chapitre ont confirmé ces propriétés. Nous avons ensuite proposé la détermination analytique des matrices d'interactions associées aux points \mathbf{P}_1 et \mathbf{P}_2 calculés à partir des nouveaux moments " shifted ". Une condition nécessaire et suffisante de stabilité a été donnée .

Chapitre 4

Validation des résultats en asservissement visuel

4.1 Introduction

Dans cette partie, nous nous intéressons à l'application des travaux de modélisation présentés dans le chapitre II pour des tâches de positionnement par asservissement visuel. Nous présentons les résultats validant le choix des informations visuelles $\mathbf{s} = (x_n \ y_n \ a_n \ r_i \ r_j \ \theta)$ pour l'asservissement visuel par rapport à un objet plan et continu. Dans les séries de simulations que nous présentons dans ce chapitre, nous avons principalement utilisé des objets continus de formes asymétriques. Nous présentons également quelques résultats très satisfaisants obtenus pour un objet symétrique. Quelque soit l'objet utilisé, les mêmes combinaisons d'invariants pour commander les mouvements de rotations le long de l'axe x et de l'axe y ont été utilisés car ce choix d'invariants ne dépend pas de l'objet considéré. Contrairement à la revue de la littérature, il fallait discuter du choix des invariants r_i et r_j selon l'objet considéré.

Le choix des couples d'invariants (r_i, r_j) à introduire dans la boucle de commande ne dépend pas de l'objet. Pour les objets considérés, il faudra choisir le meilleur couple d'invariants (r_i, r_j) parmi :

$$r_{s1} = \frac{I_{s1}}{I_{s2}^{8/10}}$$

ou

$$r_{s2} = \frac{I_{s3}}{I_{s2}^{8/10}}$$

ou

$$r_{s3} = \frac{I_{s3}}{I_{s1}}$$

qui permet d'avoir à la fois un bon conditionnement de la matrice d'interaction et un bon comportement de l'erreur $\mathbf{e}_r = (r_i - r_i^*)^2 + (r_j - r_j^*)^2$ en fonction des rotations w_x et w_y autour de la position désirée. En effet, nous cherchons à obtenir un bon comportement de l'erreur qui doit présenter un minimum global tel que la zone d'influence associée à ce minimum soit la plus large et la plus symétrique possible.

Dans la suite de ce chapitre, nous présentons des résultats sur le choix des invariants pour les trois objets donnés par les figures 4.1.a, 4.1.b et 4.1.c.

Les simulations proposées dans cette section sont réalisées sous le logiciel Matlab et sont relatives à la configuration eye-in-hand (caméra embarquée). Nous avons considéré dans ces simulations que les paramètres du plan de l'objet dans le repère caméra sont donnés approximativement pour la position désirée. Dans le cas où l'objet est parallèle au plan image pour la position désirée, nous avons fixé ($A=B=0$, $C=2$, qui correspond à $Z^*=0.5$ m). Dans la suite de ce paragraphe, nous testons d'abord notre schéma de commande dans le cas où un mouvement de rotation pur a été considéré entre la position initiale et la position désirée de la caméra. Nous intéressons ensuite au cas où les déplacements à réaliser sont composés de grandes rotations et de translations. Puis nous testons la robustesse de notre méthode vis-à-vis d'erreurs de modélisation.

Dans toutes ces simulations, la valeur de la matrice d'interaction calculée pour l'image désirée ($\widehat{\mathbf{L}}_s = \widehat{\mathbf{L}}_{s/s=s*}$) a été utilisée dans la loi de commande et la valeur du gain a été fixée à 0.2.

Dans les simulations qui suivent, nous testons d'abord les propriétés d'invariance calculées à partir des moments "shifted". A cette fin, nous avons considéré deux objets de forme différente définis par les coordonnées suivantes dans l'image :

$$\mathbf{X}_{i1} = \begin{bmatrix} -0.5 & 0.5 & 0.5 & -0.5 \\ -0.3 & -0.3 & 0.3 & 0.3 \end{bmatrix} \quad (4.1)$$

$$\mathbf{X}_{i2} = \begin{bmatrix} 0.5 & 0.5 & 0.4 & -0.65 & -0.25 \\ -0.3 & -0.3 & 0.2 & 0.35 & -0.2 \end{bmatrix} \quad (4.2)$$

Le premier objet est un rectangle (objet symétrique) représenté sur la figure 4.2(a) et le deuxième objet est un polygone non symétrique comme représenté sur la figure 4.2(b). Pour les deux objets, nous avons considéré 4 positions différentes dans l'image : la première position est définie par 4.2(a) et 4.2(b), alors que les trois autres sont obtenus en appliquant respectivement trois transformations différentes, un mouvement de translation défini par le vecteur $[0.1 ; 0.3]$, un changement d'échelle défini par le facteur d'échelle 0.9 et finalement une rotation dans le plan définie par un angle de 30 degrés. Les tableaux I et II montrent les valeurs des invariants obtenues pour l'objet symétrique pour les 4 positions indiquées ci-dessus. De ces Tableaux, nous pouvons voir que la deuxième ligne (obtenue après

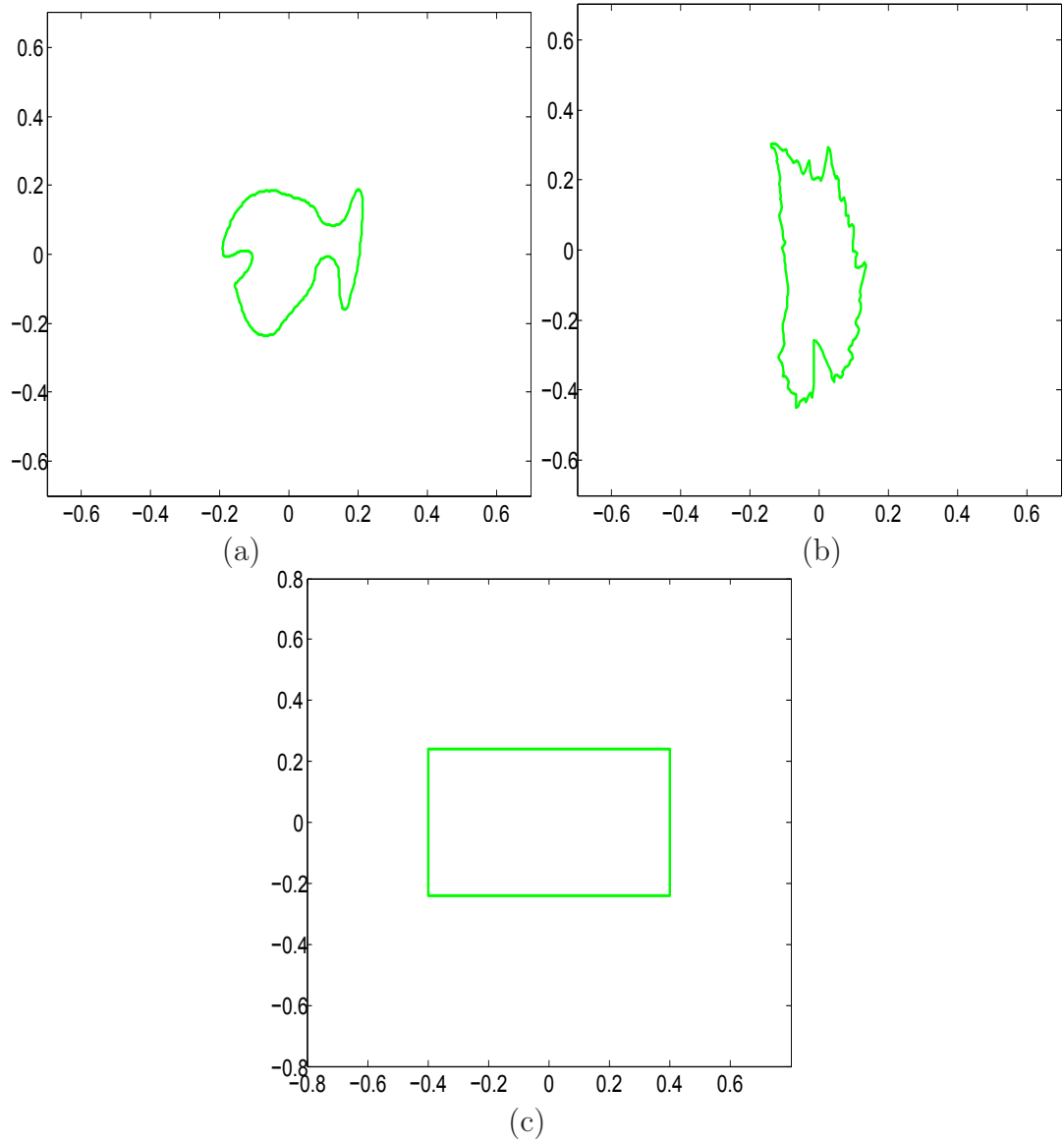


FIGURE 4.1 – Image utilisée lors de la simulation (a) image d'une baleine (b) image d'une feuille de vigne (c) image d'un rectangle

application d'une translation dans l'image) est identique à la première ligne. Ceci montre l'invariance aux translations des polynômes I_{s1} , I_{s2} et I_{s3} , mais aussi l'invariance de r_{s1} , r_{s2} et r_{s3} . A partir de la troisième ligne de chaque tableau, on peut noter qu'un changement d'échelle dans l'image modifie les valeurs de I_{s1} , I_{s2} et I_{s3} mais sans modifier les valeurs des rapports r_{s1} , r_{s2} et r_{s3} . Et finalement, la quatrième ligne de chaque tableau obtenu à partir d'une rotation par rapport à la position 1, est identique à la première ligne. Ceci montre l'invariance à la

rotation dans le plan des invariants I_{s1} , I_{s2} et I_{s3} et les rapports r_{s1} , r_{s2} et r_{s3} . Les mêmes conclusions peuvent être faites pour les résultats obtenus en utilisant un polygone asymétrique (voir les tableaux III et IV). Par conséquent, l'ensemble des résultats montrent la validité des propriétés d'invariance des primitives proposées pour commander le mouvement de rotation autour de l'axe w_x et w_y .

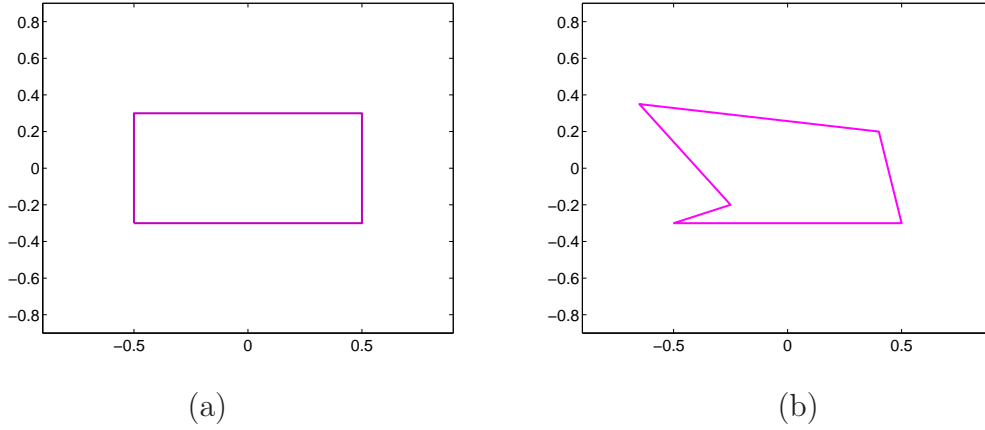


FIGURE 4.2 – Image utilisée pour prouver l'invariance (a) image d'une figure symétrique (b) image d'une figure non symétrique

Position	I_{s1}	I_{s2}	I_{s3}	r_{s1}	r_{s2}	r_{s3}
1	0.0037	-0.0013	0.028	-1.47	31.39	-0.047
2	0.0037	-0.0013	0.028	-1.47	31.39	-0.047
3	0.0016	-0.0005	0.010	-1.47	31.39	-0.047
4	0.0037	-0.0013	0.028	-1.47	31.39	-0.047

TABLE 4.1 – Valeurs des Invariants pour un objet de forme symétrique : P_1 comme point de décalage

4.2 AV2D : Rotation pure

Dans cette simulation, le déplacement est une rotation pure. La pose initiale de la caméra relativement à sa pose désirée est donnée par $\mathbf{T} = (0.0; 0.0; 0.0)$ (m) et $\theta\mathbf{u} = (-45.0; -35.0; 180)$ (Degrés), où \mathbf{T} est le vecteur définissant la translation et $\theta\mathbf{u}$ le vecteur de rotation. Une large rotation autour de l'axe optique est nécessaire pour passer de la pose initiale à la pose désirée. Pour les autres paramètres de simulations, nous avons choisi $r_{s1} = \frac{I_{s1}}{8I_{s2}^{10}}$ et comme paramètre "a" commun aux

Position	I _{s1}	I _{s2}	I _{s3}	r _{s1}	r _{s2}	r _{s3}
1	0.0087	-0.0021	0.0198	-0.78	7.42	-0.10
2	0.0087	-0.0021	0.0198	-0.78	7.42	-0.10
3	0.0038	-0.0007	0.0069	-0.78	7.42	-0.10
4	0.0087	-0.0021	0.0198	-0.78	7.42	-0.10

TABLE 4.2 – Valeurs des Invariants pour un objet de forme symétrique : P_2 comme point de décalage

Position	I _{s1}	I _{s2}	I _{s3}	r _{s1}	r _{s2}	r _{s3}
1	0.0017	-0.0006	0.0134	-1.59	37.8	-0.042
2	0.0017	-0.0006	0.0134	-1.59	37.8	-0.042
3	0.0007	-0.0002	0.0047	-1.59	37.8	-0.042
4	0.0017	-0.0006	0.0134	-1.59	37.8	-0.042

TABLE 4.3 – Valeurs des Invariants pour un objet de forme non symétrique : P_1 comme point de décalage

deux points \mathbf{P}_1 et \mathbf{P}_2 nous avons utilisé $a = (\mu_{20} + \mu_{02})^{\frac{1}{4}}$.

4.2.1 Cas de la Baleine

Le choix des informations visuelles $\mathbf{s} = (x_n \ y_n \ a_n \ r_i \ r_j \ \theta)$ pour la valeur initiale et la valeur désirée de \mathbf{s} est donc

$$\mathbf{s}_i = (-0.12 \ 0.12 \ 2.00 \ -1.12 \ -1.17 \ 0.95)$$

$$\mathbf{s}^* = (0.00 \ 0.00 \ 2.00 \ -1.11 \ -1.17 \ 0.70)$$

On note donc une orientation très différente de la pose désirée suivant l'axe optique. Le calcul de "a" donne la valeur $a = 0.2144$ pour la pose désirée. Ceci nous permet donc de déduire les coordonnées des points \mathbf{P}_1 et \mathbf{P}_2 comme suit :

Pour la valeur désirée \mathbf{P}_1^* a pour coordonnées :

$$x_{sh1}^* = 0.1637$$

$$y_{sh1}^* = 0.1385$$

Pour la valeur désirée \mathbf{P}_2^* a pour coordonnées :

$$x_{sh2}^* = -0.1385$$

$$y_{sh2}^* = 0.1637$$

Les coordonnées du centre de gravité de l'image à la position désirée sont données par :

$$x_g = 0.000 \text{ et } y_g = 0.000$$

$$\text{Pour la valeur initiale } a = 0.2152$$

Position	I_{s1}	I_{s2}	I_{s3}	r_{s1}	r_{s2}	r_{s3}
1	0.0046	-0.0009	0.0083	-0.7327	6.86	-0.1068
2	0.0046	-0.0009	0.0083	-0.7327	6.86	-0.1068
3	0.0020	-0.0003	0.0029	-0.7327	6.86	-0.1068
4	0.0046	-0.0009	0.0083	-0.7327	6.86	-0.1068

TABLE 4.4 – Valeurs des Invariants pour un objet de forme non symétrique : P_2 comme point de décalage

\mathbf{P}_1 a pour coordonnées :

$$xsh1 = 0.1258$$

$$ysh1 = 0.1746$$

\mathbf{P}_2 a pour coordonnées :

$$xsh2 = -0.1746$$

$$ysh2 = 0.1258$$

Les coordonnées du centre de gravité de l'image à la position initiale sont données par :

$$xg = -0.0578 \text{ et } yg = 0.0579$$

Les résultats obtenus sont présentés dans les figures 4.3.

4.2.2 Cas de la Feuille de vigne

Le choix des informations visuelles $\mathbf{s} = (x_n \ y_n \ a_n \ r_i \ r_j \ \theta)$ pour la valeur initiale et la valeur désirée de \mathbf{s} est donc

$$\mathbf{s}_i = (0.12 \ -0.01 \ 2.50 \ -0.58 \ -2.09 \ -1.27)$$

$$\mathbf{s}^* = (0.02 \ -0.17 \ 2.50 \ -0.56 \ -2.07 \ -1.53)$$

On note donc une orientation très différente de la pose désirée suivant l'axe optique.

Le calcul de "a" donne la valeur $a = 0.2554$ pour la pose désirée. Ceci nous permet donc de déduire les coordonnées des points \mathbf{P}_1 et \mathbf{P}_2 comme suit :

Pour la valeur désirée \mathbf{P}_1^* a pour coordonnées :

$$xsh1^* = 0.0106$$

$$ysh1^* = -0.2552$$

Pour la valeur désirée \mathbf{P}_2^* a pour coordonnées :

$$xsh2^* = 0.2552$$

$$ysh2^* = 0.0106$$

Les coordonnées du centre de gravité de l'image à la position désirée sont données par :

xg = 0.000 et yg = 0.000

Pour la valeur initiale a = 0.2552

\mathbf{P}_1 a pour coordonnées :

xsh1 = 0.0756

ysh1 = -0.2437

\mathbf{P}_2 a pour coordonnées :

xsh2 = 0.2437

ysh2 = 0.0756

Les coordonnées du centre de gravité de l'image à la position initiale sont données par :

xg = -0.0487 et yg = -0.0049

Les résultats obtenus sont présentés dans les figures 4.4.

4.2.3 Cas du Rectangle

Les coordonnées des quatre points du rectangle exprimées dans le repère de la caméra pour la pose initiale et pour la pose désirée sont respectivement données par :

$$\begin{aligned} {}^cP_1 &= \begin{bmatrix} -0.385 & -0.277 & 1.000 \end{bmatrix}^T \\ {}^cP_2 &= \begin{bmatrix} 0.379 & -0.075 & 1.000 \end{bmatrix}^T \\ {}^cP_3 &= \begin{bmatrix} 0.273 & 0.388 & 1.000 \end{bmatrix}^T \\ {}^cP_4 &= \begin{bmatrix} -0.520 & 0.195 & 1.000 \end{bmatrix}^T \\ {}^cP_1^* &= \begin{bmatrix} -0.400 & -0.240 & 1.000 \end{bmatrix}^T \\ {}^cP_2^* &= \begin{bmatrix} 0.400 & -0.240 & 1.000 \end{bmatrix}^T \\ {}^cP_3^* &= \begin{bmatrix} 0.400 & 0.240 & 1.000 \end{bmatrix}^T \\ {}^cP_4^* &= \begin{bmatrix} -0.400 & 0.240 & 1.000 \end{bmatrix}^T \end{aligned}$$

Le choix des informations visuelles $\mathbf{s} = (x_n \ y_n \ a_n \ r_i \ r_j \ \theta)$ pour la valeur initiale et la valeur désirée de \mathbf{s} est donc

$$\mathbf{s}_i = (-0.16 \ 0.15 \ 2.50 \ -0.78 \ -1.49 \ 0.25)$$

$$\mathbf{s}^* = (0.00 \ 0.00 \ 2.50 \ -0.79 \ -1.48 \ 0.00)$$

On note donc une orientation très différente de la pose désirée suivant l'axe optique.

Le calcul de "a" donne la valeur a = 0.4085 pour la pose désirée. Ceci nous permet donc de déduire les coordonnées des points P1 et P2 comme suit :

Pour la valeur désirée \mathbf{P}_1^* a pour coordonnées :

$$xsh1^* = 0.4085$$

$$ysh1^* = 0.0000$$

Pour la valeur désirée \mathbf{P}_2^* a pour coordonnées :

$$xsh2^* = 0.0000$$

$$ysh2^* = 0.4085$$

Les coordonnées du centre de gravité de l'image à la position désirée sont données par :

$$xg = 0.000 \text{ et } yg = 0.000$$

Pour la valeur initiale $a = 0.4106$

\mathbf{P}_1 a pour coordonnées :

$$xsh1 = 0.3983$$

$$ysh1 = 0.0998$$

\mathbf{P}_2 a pour coordonnées :

$$xsh2 = -0.0998$$

$$ysh2 = 0.3983$$

Les coordonnées du centre de gravité de l'image à la position initiale sont données par :

$$xg = -0.0656 \text{ et } yg = 0.0588$$

Les résultats obtenus sont présentés dans les figures 4.5.

4.3 AV2D : Déplacement générique

Dans cette simulation le mouvement est donc complexe. Le déplacement est composé d'un mouvement de translation et d'un mouvement de rotation. Pour les autres paramètres de simulations, nous avons choisi $r_{s1} = \frac{I_{s1}}{8 I_{s2}^{10}}$ et comme para-

mètre "a" commun aux deux points \mathbf{P}_1 et \mathbf{P}_2 nous avons utilisé $a = (\mu_{20} + \mu_{02})^{\frac{1}{4}}$.

4.3.1 Cas de la Baleine

La pose initiale de la caméra relativement à sa pose désirée est donnée par $\mathbf{T} = (-1.0; 1.5; 0.8)$ (m) et $\theta\mathbf{u} = (25.0; 20.0; -50)$ (Degrés), où \mathbf{T} est le vecteur définissant la translation et $\theta\mathbf{u}$ le vecteur de rotation. Une large translation est nécessaire pour passer de la pose initiale à la pose désirée.

Le choix des informations visuelles $\mathbf{s} = (x_n \ y_n \ a_n \ r_i \ r_j \ \theta)$ pour la valeur initiale et la valeur désirée de \mathbf{s} est donc

$$\mathbf{s}_i = (-0.89 \ 1.32 \ 2.00 \ -1.03 \ -1.12 \ 0.51)$$

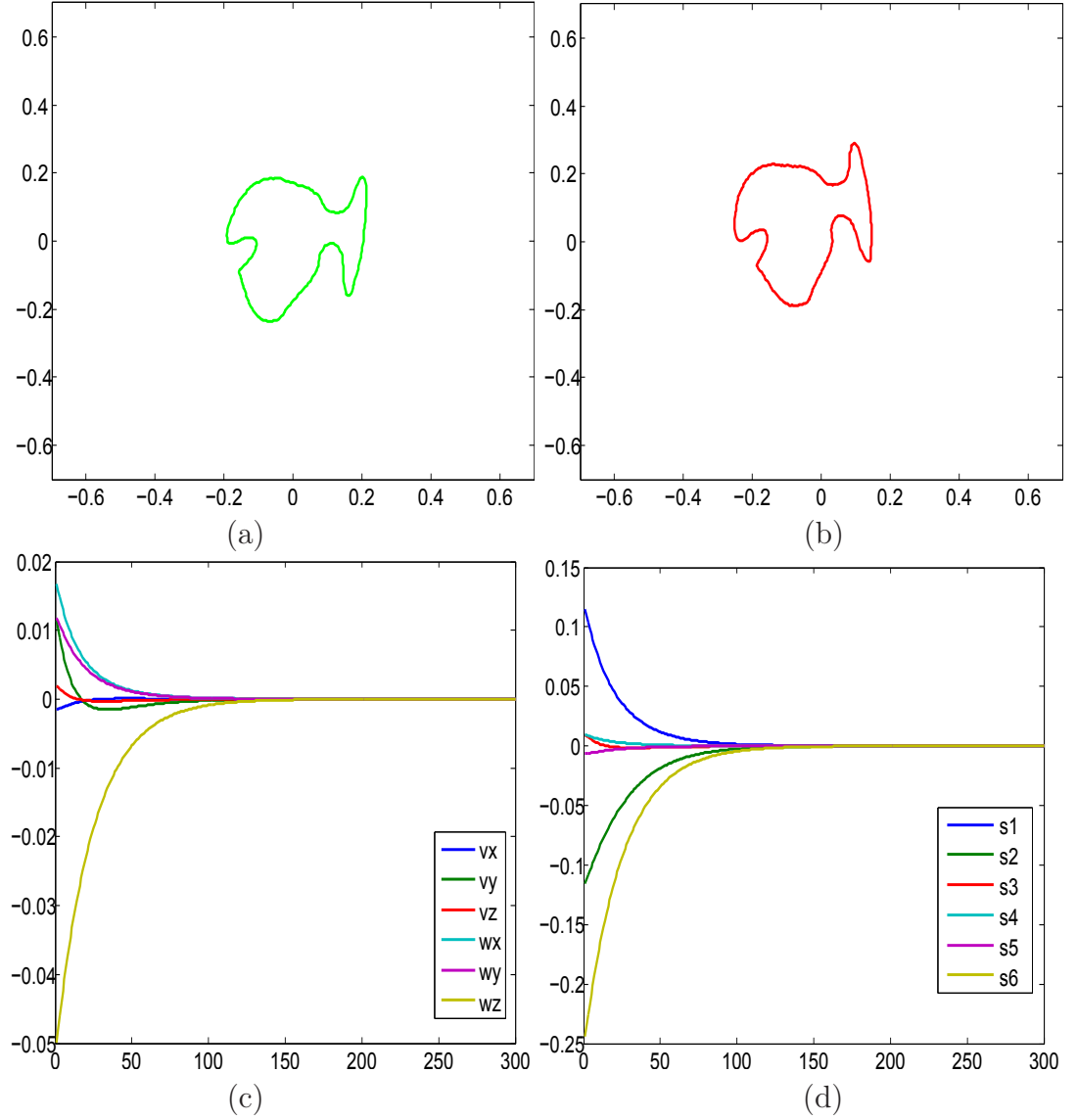


FIGURE 4.3 – Résultats de Simulation pour un mouvement de rotation pure : image d'une baleine. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".

$$\mathbf{s}^* = \begin{pmatrix} 0.00 & 0.00 & 2.00 & -1.11 & -1.17 & 0.70 \end{pmatrix}$$

Comme nous le constatons, les déplacements effectués sont importants. L'objet est éloigné de sa position désirée.

Le calcul de "a" donne la valeur $a = 0.2144$ pour la pose désirée. Ceci nous permet donc de déduire les coordonnées des points \mathbf{P}_1 et \mathbf{P}_2 comme suit :

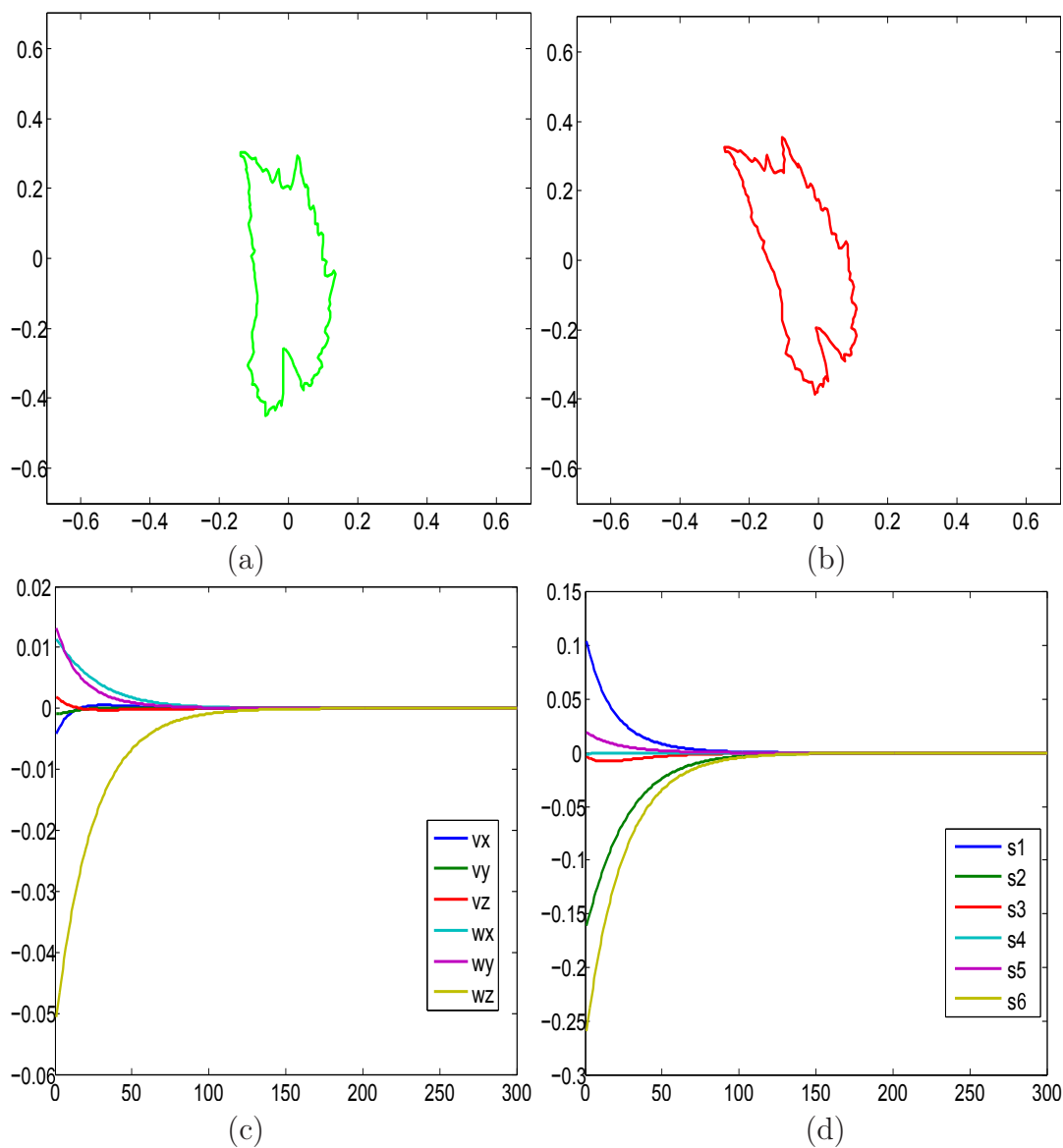


FIGURE 4.4 – Résultats de Simulation pour un mouvement de rotation pure : image d'une feuille de vigne. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".

Pour la valeur désirée \mathbf{P}_1^* a pour coordonnées :

$$xsh1^* = 0.1637$$

$$ysh1^* = 0.1385$$

Pour la valeur désirée \mathbf{P}_2^* a pour coordonnées :

$$xsh2^* = -0.1385$$

$$ysh2^* = 0.1637$$

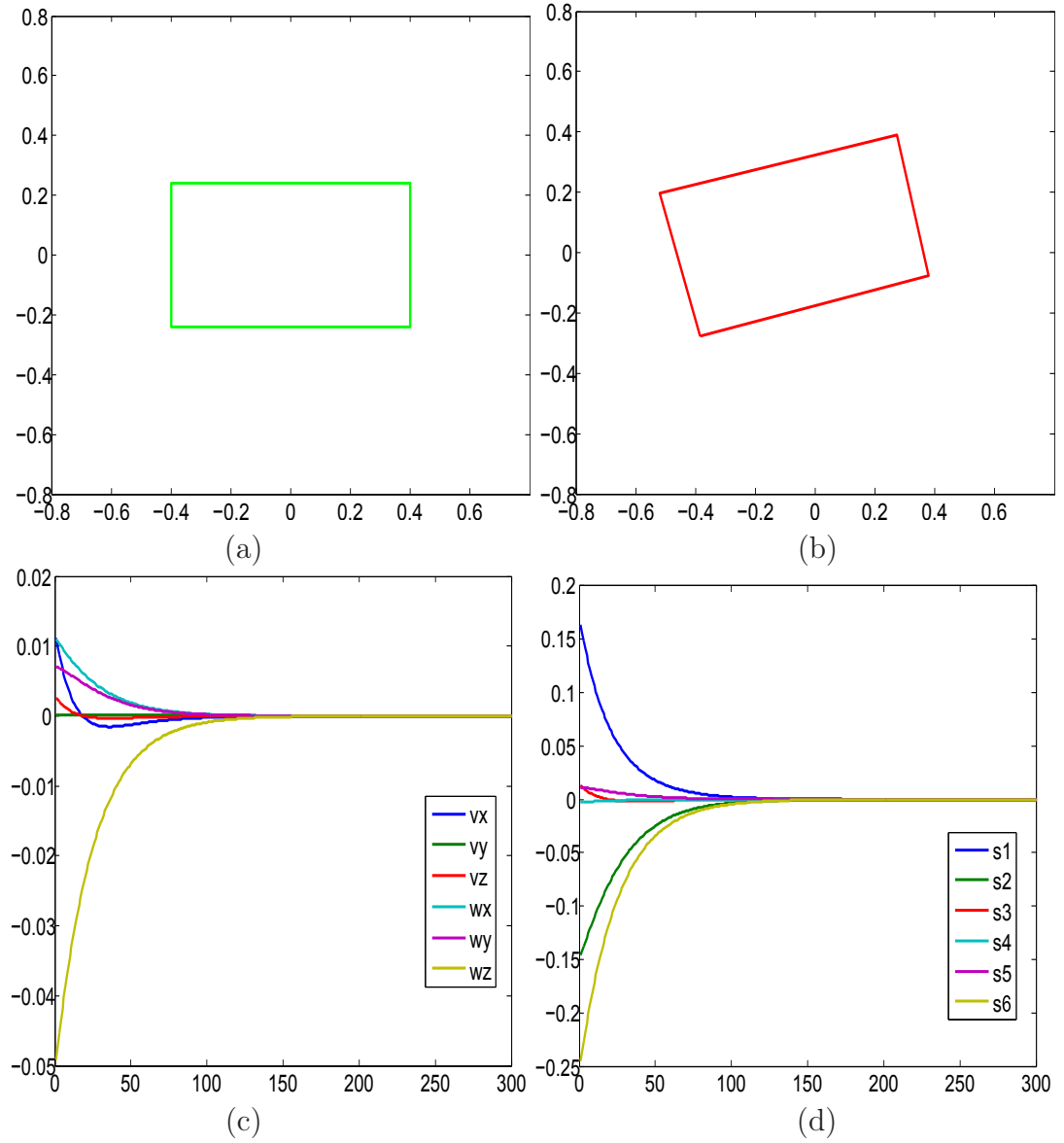


FIGURE 4.5 – Résultats de Simulation pour un mouvement de rotation pure : image d'un rectangle. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".

Les coordonnées du centre de gravité de l'image à la position désirée sont données par :

$$x_g = 0.000 \text{ et } y_g = 0.000$$

Pour la valeur initiale $a = 0.1483$

\mathbf{P}_1 a pour coordonnées :

$$x_{sh1} = 0.1294$$

$$ysh1 = 0.0726$$

\mathbf{P}_2 a pour coordonnées :

$$xsh2 = -0.0726$$

$$ysh2 = 0.1294$$

Les coordonnées du centre de gravité de l'image à la position initiale sont données par :

$$xg = -0.3044 \text{ et } yg = 0.4522$$

Les résultats obtenus sont présentés dans les figures 4.6.

4.3.2 Cas de la Feuille de vigne

La pose initiale de la caméra relativement à sa pose désirée est donnée par $\mathbf{T} = (1.8; -0.9; 0.6)$ (m) et $\theta\mathbf{u} = (-25.0; -25.0; 60.0)$ (Degrés), où \mathbf{T} est le vecteur définissant la translation et $\theta\mathbf{u}$ le vecteur de rotation. Une large translation est nécessaire pour passer de la pose initiale à la pose désirée.

Le choix des informations visuelles $\mathbf{s} = (x_n \ y_n \ a_n \ r_i \ r_j \ \theta)$ pour la valeur initiale et la valeur désirée de \mathbf{s} est donc

$$\mathbf{s}_i = (1.62 \ -0.88 \ 3.23 \ -0.58 \ -2.12 \ -1.36)$$

$$\mathbf{s}^* = (0.02 \ -0.17 \ 2.50 \ -0.56 \ -2.07 \ -1.53)$$

Comme nous le constatons, les déplacements effectués sont importants. L'objet est éloigné de sa position désirée.

Le calcul de "a" donne la valeur $a = 0.2554$ pour la pose désirée. Ceci nous permet donc de déduire les coordonnées des points \mathbf{P}_1 et \mathbf{P}_2 comme suit :

Pour la valeur désirée \mathbf{P}_1^* a pour coordonnées :

$$xsh1^* = 0.0106$$

$$ysh1^* = -0.2552$$

Pour la valeur désirée \mathbf{P}_2^* a pour coordonnées :

$$xsh2^* = 0.2552$$

$$ysh2^* = 0.0106$$

Les coordonnées du centre de gravité de l'image à la position désirée sont données par :

$$xg = 0.000 \text{ et } yg = 0.000$$

Pour la valeur initiale $a = 0.1976$

\mathbf{P}_1 a pour coordonnées :

$$xsh1 = 0.0419$$

$$ysh1 = -0.1931$$

\mathbf{P}_2 a pour coordonnées :

$$xsh2 = 0.1931$$

$$ysh2 = 0.0419$$

Les coordonnées du centre de gravité de l'image à la position initiale sont données par :

$$xg = 0.5022 \text{ et } yg = -0.2719$$

Les résultats obtenus sont présentés dans les figures 4.7.

4.3.3 Cas du Rectangle

La pose initiale de la caméra relativement à sa pose désirée est donnée par $\mathbf{T} = (1.8; -1.5; 0.6)$ (m) et $\theta\mathbf{u} = (-25.0; -25.0; 60.0)$ (Degrés), où \mathbf{T} est le vecteur définissant la translation et $\theta\mathbf{u}$ le vecteur de rotation. Une large translation est nécessaire pour passer de la pose initiale à la pose désirée.

Les coordonnées des quatre points du rectangle exprimées dans le repère de la caméra pour la pose initiale et pour la pose désirée sont respectivement données par :

$$\begin{aligned} {}^cP_1 &= \begin{bmatrix} -0.229 & -0.688 & 1.000 \end{bmatrix}^T \\ {}^cP_2 &= \begin{bmatrix} 0.820 & -0.512 & 1.000 \end{bmatrix}^T \\ {}^cP_3 &= \begin{bmatrix} 0.768 & -0.156 & 1.000 \end{bmatrix}^T \\ {}^cP_4 &= \begin{bmatrix} 0.150 & -0.320 & 1.000 \end{bmatrix}^T \\ {}^cP_1^* &= \begin{bmatrix} -0.400 & -0.240 & 1.000 \end{bmatrix}^T \\ {}^cP_2^* &= \begin{bmatrix} 0.400 & -0.240 & 1.000 \end{bmatrix}^T \\ {}^cP_3^* &= \begin{bmatrix} 0.400 & 0.240 & 1.000 \end{bmatrix}^T \\ {}^cP_4^* &= \begin{bmatrix} -0.400 & 0.240 & 1.000 \end{bmatrix}^T \end{aligned}$$

Le choix des informations visuelles $\mathbf{s} = (x_n \ y_n \ a_n \ r_i \ r_j \ \theta)$ pour la valeur initiale et la valeur désirée de \mathbf{s} est donc

$$\mathbf{s}_i = (1.58 \ -1.35 \ 3.23 \ -0.77 \ -1.53 \ 0.32)$$

$$\mathbf{s}^* = (0.00 \ 0.00 \ 2.50 \ -0.79 \ -1.48 \ 0.00)$$

Comme nous le constatons, les déplacements effectués sont importants. L'objet est éloigné de sa position désirée.

Le calcul de "a" donne la valeur $a = 0.4085$ pour la pose désirée. Ceci nous permet donc de déduire les coordonnées des points \mathbf{P}_1 et \mathbf{P}_2 comme suit :

Pour la valeur désirée \mathbf{P}_1^* a pour coordonnées :

$$xsh1^* = 0.4085$$

$$ysh1^* = 0.0000$$

Pour la valeur désirée \mathbf{P}_2^* a pour coordonnées :

$$xsh2^* = 0.0000$$

$$ysh2^* = 0.4085$$

Les coordonnées du centre de gravité de l'image à la position désirée sont données par :

$$xg = 0.000 \text{ et } yg = 0.000$$

Pour la valeur initiale $a = 0.3173$

\mathbf{P}_1 a pour coordonnées :

$$xsh1 = 0.3010$$

$$ysh1 = 0.1005$$

\mathbf{P}_2 a pour coordonnées :

$$xsh2 = -0.1005$$

$$ysh2 = 0.3010$$

Les coordonnées du centre de gravité de l'image à la position initiale sont données par :

$$xg = 0.4891 \text{ et } yg = -0.4186$$

Les résultats obtenus sont présentés dans les figures 4.8.

Les valeurs numériques de la matrice d'interaction calculée pour la position désirée pour chaque objet sont respectivement :

$$\mathbf{L}_{s/s^*}^{\parallel} = \begin{bmatrix} -1 & 0 & 0 & 0.01 & -2.08 & 0 \\ 0 & -1 & 0 & 2.08 & -0.01 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.05 & -0.10 & 0 \\ 0 & 0 & 0 & -0.13 & 0.29 & 0 \\ 0 & 0 & 0 & -0.45 & 0.55 & -1 \end{bmatrix} \quad (4.3)$$

$$\mathbf{L}_{s/s^*}^{\parallel} = \begin{bmatrix} -1 & 0 & 0 & 0.01 & -2.54 & -0.17 \\ 0 & -1 & 0 & 2.80 & 0.01 & 0.02 \\ 0 & 0 & -1 & 0.26 & -0.03 & 0 \\ 0 & 0 & 0 & -0.01 & 0.03 & 0 \\ 0 & 0 & 0 & -0.16 & -0.15 & 0 \\ 0 & 0 & 0 & 0.12 & -0.01 & -1 \end{bmatrix} \quad (4.4)$$

$$\mathbf{L}_{s/s^*}^{\parallel} = \begin{bmatrix} -1 & 0 & 0 & 0 & -3.03 & 0 \\ 0 & -1 & 0 & 2.69 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.32 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad (4.5)$$

L'analyse de ces différentes matrices d'interactions nous prouve l'invariance à la translation, à la rotation et à l'échelle des primitives visuelles ri et rj calculés à partir des moments géométriques "shifted".

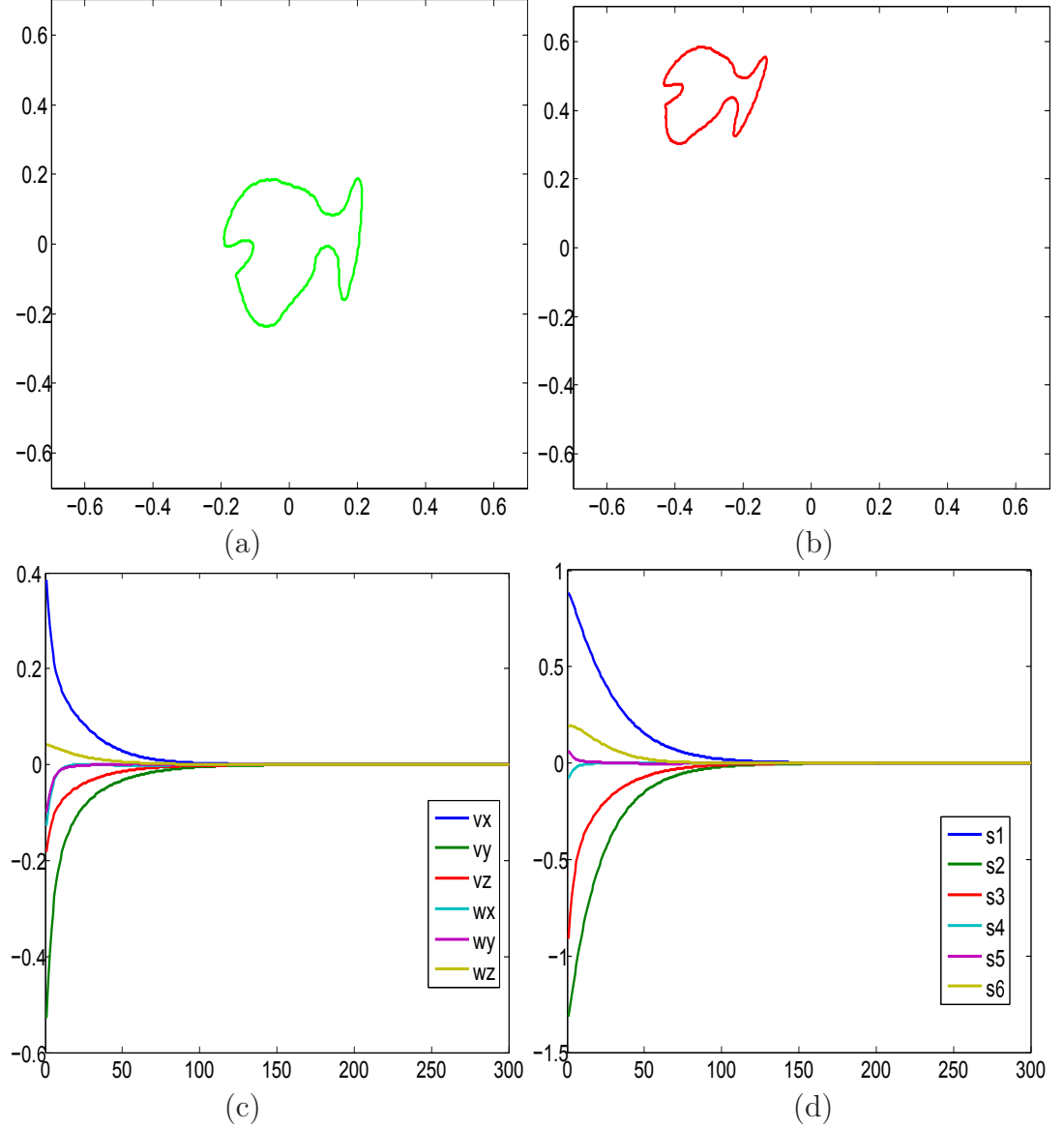


FIGURE 4.6 – Résultats de Simulation pour un mouvement générique : image d'une baleine. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".

4.4 Analyse des résultats et Discussion

4.4.1 Analyse des résultats

En utilisant une 'Baleine' comme objet, les erreurs de suivi ($\mathbf{s}(t) - \mathbf{s}^*(t)$) restent suffisamment faibles durant toute la phase d'asservissement pour assurer le

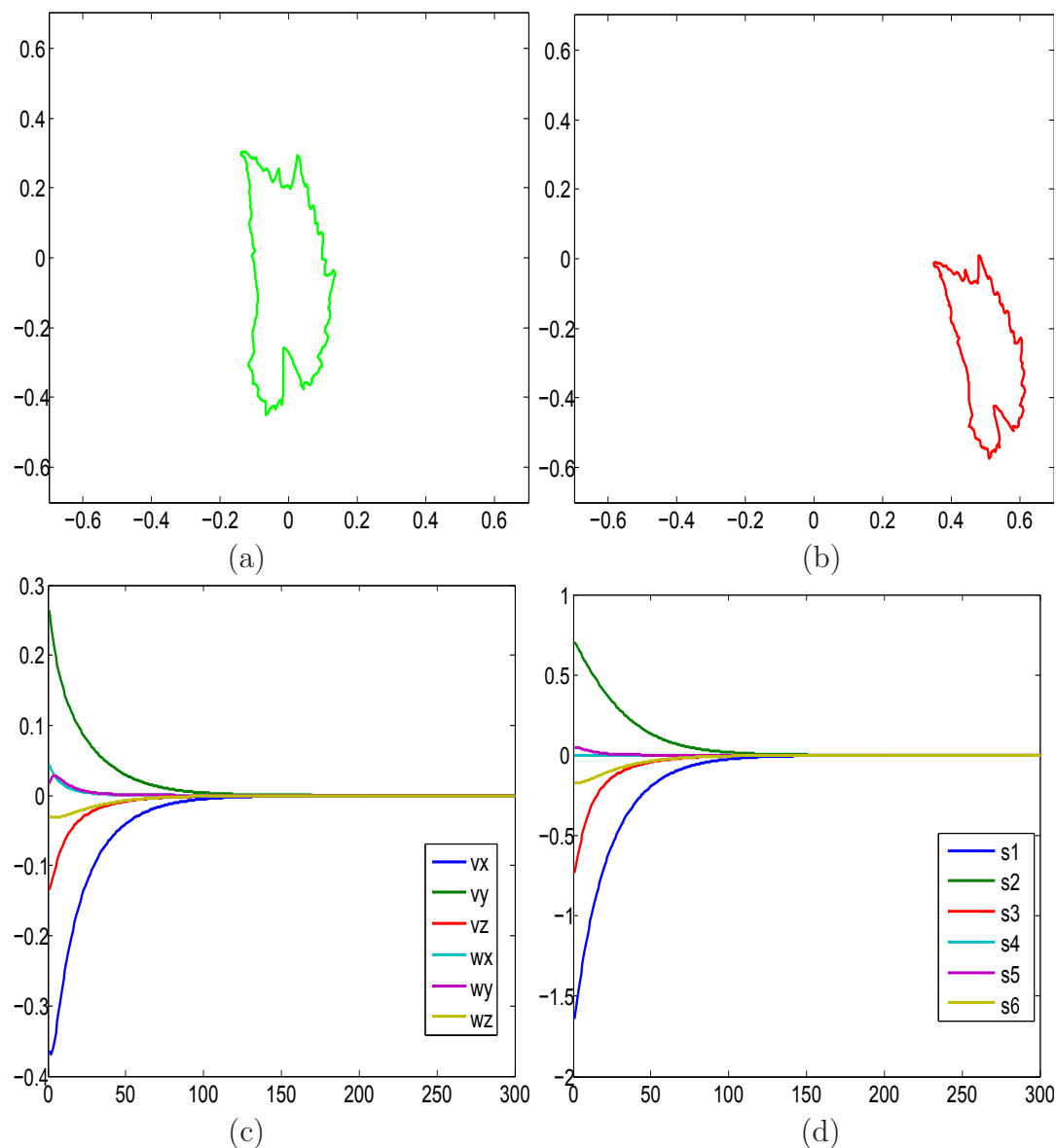


FIGURE 4.7 – Résultats de Simulation pour un mouvement générique : image d'une feuille de vigne. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".

bon comportement de la loi de commande 2D. La convergence des informations visuelles considérées vers leur valeur désirée démontre que la tâche a été correctement réalisée. Les variations des vitesses de translation et de rotation de la caméra sont très satisfaisantes en raison de la régularité de la fonction de tâche. Les courbes associées aux différentes composantes des vitesses de translation et de rotation montrent clairement un découplage entre elles. Les différents résultats

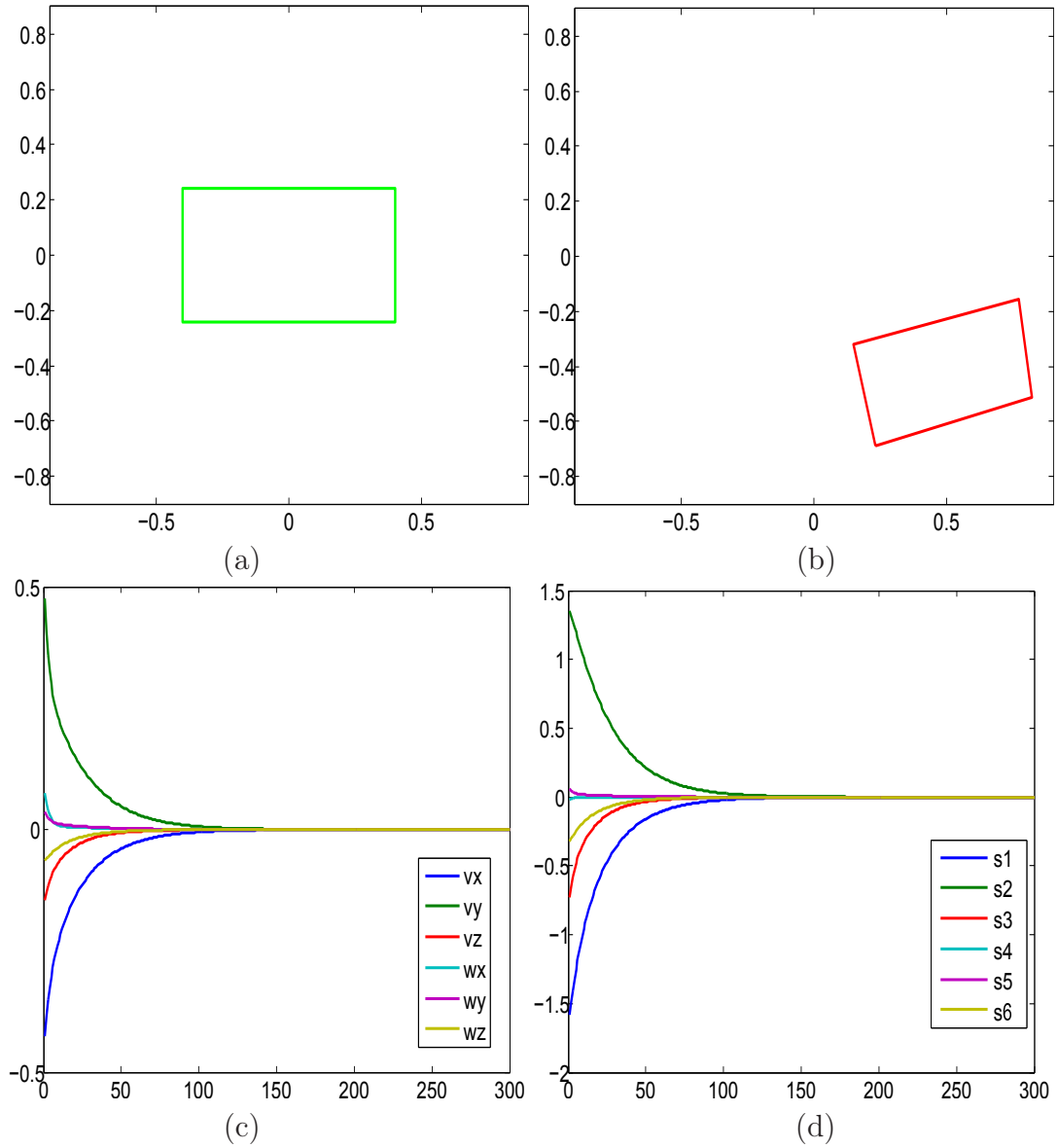


FIGURE 4.8 – Résultats de Simulation pour un mouvement générique : image d'un rectangle. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".

montrent que le positionnement est bien réalisé. Les mêmes conclusions peuvent être faites pour les résultats obtenus en utilisant comme objet 'une feuille de Vigne' et un 'Rectangle'.

4.4.2 Discussion

Nous avons validé dans cette première partie les corrections que nous avons apportées à la forme analytique de la matrice d'interaction dans le cas des objets complexes. Le cas d'un objet de forme symétrique a aussi été étudié. Nous avons constaté que l'utilisation des moments "shifted" nous permet de s'affranchir de la forme de l'objet pour contrôler les rotations autour des axes x et y qui posait d'énormes problèmes en asservissement visuel et surtout quand l'objet utilisé est de forme symétrique. Comme la simulation est réalisée dans le cas parfait, nous validons ensuite la robustesse de la commande aux erreurs de modélisation en ajoutant du bruit aux mesures.

4.5 Résultats avec des erreurs de modélisation

Les simulations décrites jusqu'à présent considèrent un modèle idéal du processus contrôlé. Dans le cas réel, il est impossible d'obtenir un modèle qui représente exactement le système. Différentes sources d'incertitude font que le modèle utilisé diffère légèrement du processus. De plus, tous les processus subissent des perturbations qui peuvent modifier leur comportement par rapport à la phase d'identification des paramètres qui servent dans le modèle de commande. Enfin, les mesures fournies par les capteurs sont toujours entachées de bruit qu'il faut prendre en considération pour assurer la stabilité du système. Pour évaluer le comportement de la commande en fonction d'un modèle cinématique erroné, une perturbation est additionnée à l'acquisition des points désirés par la caméra c'est à dire aux points courants sous la forme d'un bruit uniforme d'amplitude 0.01. Cette perturbation permet de considérer à la fois des incertitudes sur la partie dynamique du système et des perturbations sur l'état.

Nous testons maintenant la robustesse de notre méthode vis-à-vis d'erreurs de modélisation dans l'optique d'une validation future de notre choix des primitives visuelles \mathbf{s} en pratique. Dans cette simulation, un bruit blanc gaussien de 1% est introduit sur les points. La probabilité qu'un bruit blanc gaussien apparaisse sur les pixels d'un point de coordonnées (x,y) est donnée par :

$$\mathbf{p}(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp \left[- \left(\frac{(x - x_g)^2}{2\sigma_x} + \frac{(y - y_g)^2}{2\sigma_y} \right) \right] \quad (4.6)$$

Où x_g et y_g sont les coordonnées du centre de gravité de x et de y. σ_x et σ_y représentent la variance suivant x et y respectivement.

Nous présentons ici uniquement les résultats proches de la réalité. Les résultats obtenus sont donnés sur les figures 4.9, 4.10 et 4.11. Le comportement du système est similaire à celui obtenu dans le cas sans erreur de modélisation. Dans ces

conditions, l'asservissement visuel 2D fournit de très bons résultats et fait preuve d'une grande robustesse. La tâche de positionnement est correctement réalisée comme le démontre la convergence vers 0 de l'erreur des informations visuelles ($\mathbf{s}(t) - \mathbf{s}^*(t)$). Les variations des vitesses de translation et de rotation de la caméra sont très satisfaisantes. Cela montre l'efficacité du schéma de commande. À travers toute cette analyse, la robustesse de notre schéma de commande vis-à-vis de ces erreurs de modélisation est donc validée.

4.6 AV2D : Déplacement générique avec comme paramètres de décalage "a₁" et "a₂"

Dans cette simulation le mouvement est aussi complexe. Le déplacement est composé d'un mouvement de translation et d'un mouvement de rotation. Pour les autres paramètres de simulations, nous avons choisi $r_{s1} = \frac{I_{s1}}{8^{1/10}}$ et comme paramètre de décalage, nous associons au point \mathbf{P}_1 le paramètre "a₁" et au point \mathbf{P}_2 le paramètre "a₂". Pour la baleine, la pose initiale de la caméra relativement à sa pose désirée est donnée par $\mathbf{T} = (-1.0; 1.5; 0.8)$ (m) et $\theta\mathbf{u} = (25.0; 20.0; -50)$ (Degrés); pour la feuille de vine, la pose initiale de la caméra relativement à sa pose désirée est donnée par $\mathbf{T} = (1.8; -0.9; 0.6)$ (m) et $\theta\mathbf{u} = (-25.0; -25.0; 60.0)$ (Degrés); et enfin pour le rectangle, la pose initiale de la caméra relativement à sa pose désirée est donnée par $\mathbf{T} = (1.8; -1.5; 0.6)$ (m) et $\theta\mathbf{u} = (-25.0; -25.0; 60.0)$ (Degrés), où \mathbf{T} est le vecteur définissant la translation et $\theta\mathbf{u}$ le vecteur de rotation. Les résultats (voir figures 4.12, 4.13 et 4.14) obtenus montrent que le système converge et que les paramètres "a₁" et "a₂" peuvent aussi très bien être utilisés car toutes les propriétés d'invariance sont préservées. Les erreurs de suivi ($\mathbf{s}(t) - \mathbf{s}^*(t)$) restent suffisamment faibles durant toute la phase d'asservissement pour assurer le bon comportement de la loi de commande 2D.

4.7 AV2D : Déplacement générique avec comme invariant r_{s3}

Dans cette dernière simulation le mouvement est aussi complexe. Le déplacement est composé d'un mouvement de translation et d'un mouvement de rotation. Pour les autres paramètres de simulations, nous avons choisi $r_{s3} = \frac{I_{s3}}{I_{s1}}$ et comme paramètre "a" commun aux deux points \mathbf{P}_1 et \mathbf{P}_2 nous avons utilisé $a = (\mu_{20} + \mu_{02})^{\frac{1}{4}}$. Pour la baleine, la pose initiale de la caméra relativement à sa pose désirée est donnée par $\mathbf{T} = (-1.0; 1.5; 0.8)$ (m) et $\theta\mathbf{u} = (25.0; 20.0; -50)$ (Degrés); pour la feuille de vine, la pose initiale de la caméra relativement à sa

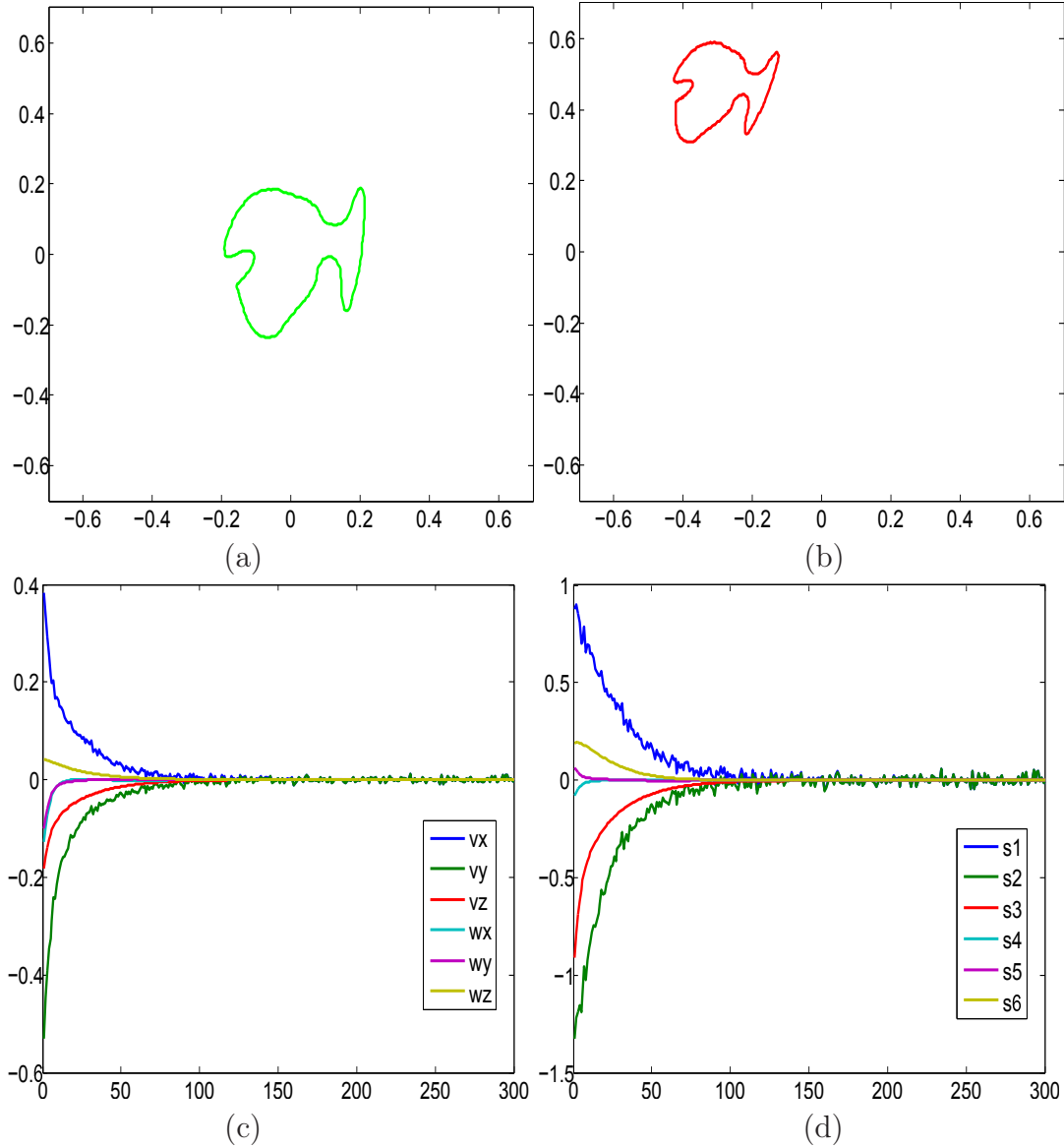


FIGURE 4.9 – Résultats de Simulation pour un mouvement générique en présence du bruit : image d'une baleine. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".

pose désirée est donnée par $\mathbf{T} = (1.8; -0.9; 0.6)$ (m) et $\theta\mathbf{u} = (-25.0; -25.0; 60.0)$ (Degrés); et enfin pour le rectangle, la pose initiale de la caméra relativement à sa pose désirée est donnée par $\mathbf{T} = (0.9; -0.5; 0.8)$ (m) et $\theta\mathbf{u} = (-45.0; -25.0; 60.0)$ (Degrés), où \mathbf{T} est le vecteur définissant la translation et $\theta\mathbf{u}$ le vecteur de rotation. Les résultats (voir figures 4.15 , 4.16 et 4.17) obtenus montrent que le système converge aussi bien.

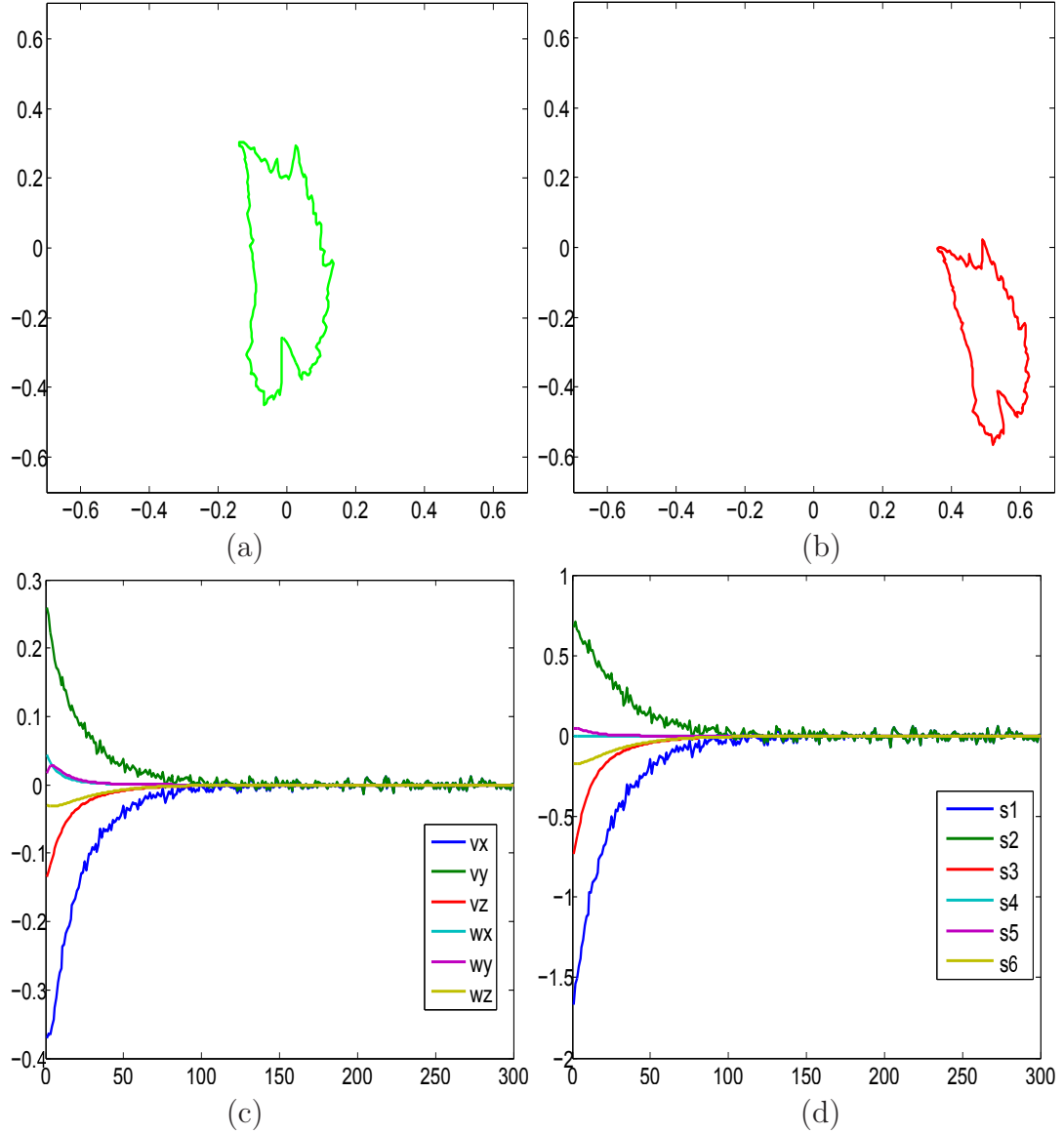


FIGURE 4.10 – Résultats de Simulation pour un mouvement générique en présence du bruit : image d'une feuille de vigne. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".

4.8 Conclusion

Dans ce chapitre, nous avons validé les résultats théoriques sur le choix des primitives visuelles obtenus au chapitre 2 en utilisant les moments "shifted". Les résultats de simulations ont été réalisés dans le cas idéal et dans un autre cas en présence du bruit. Cette commande est bien adaptée pour contrôler le dépla-

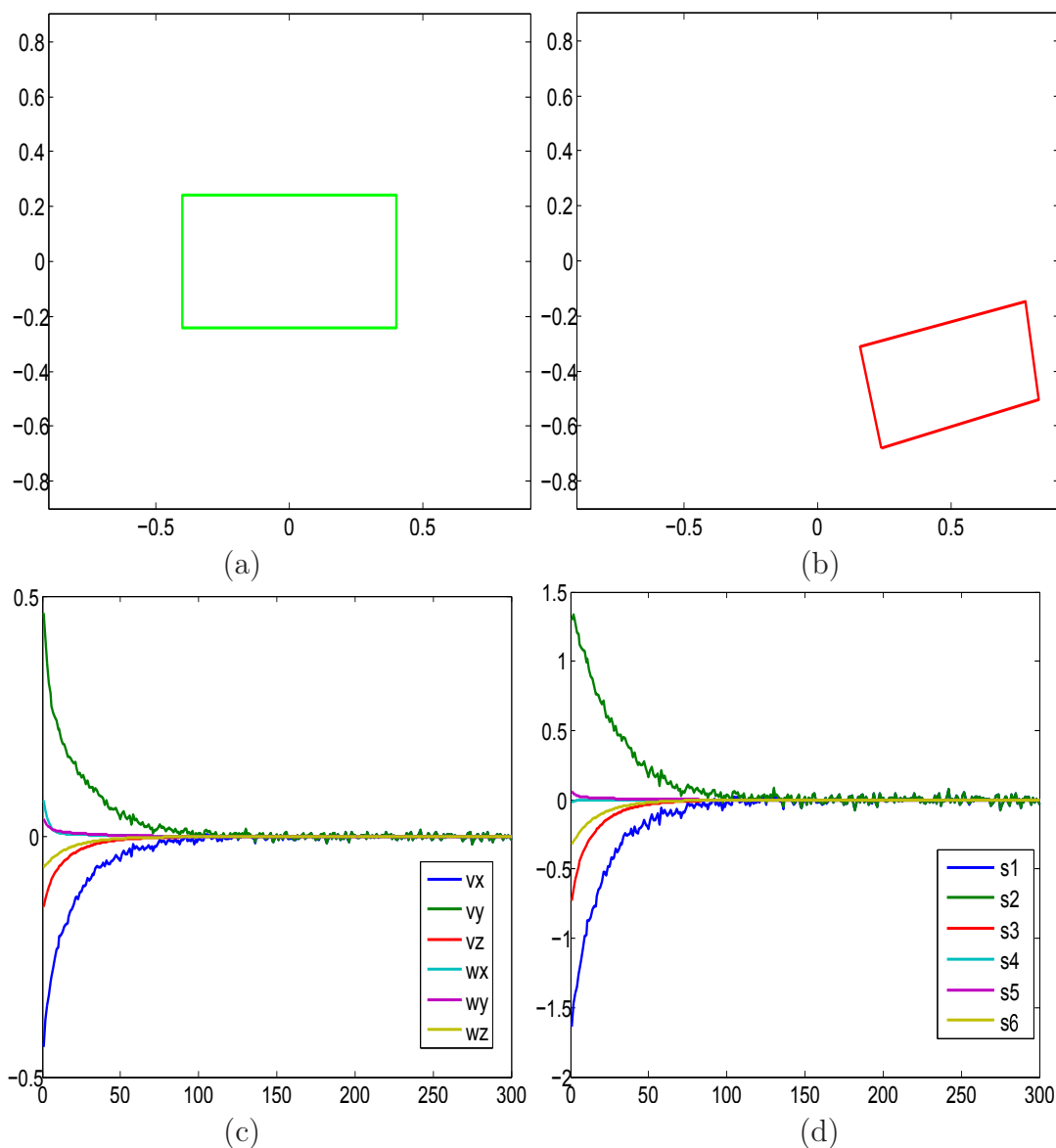


FIGURE 4.11 – Résultats de Simulation pour un mouvement générique en présence du bruit : image d'un rectangle. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".

cement du robot aussi bien dans le cas des mouvements simples que complexes et variés. Cette commande peut être appliquée à toute configuration d'objet représenté par un contour fermé. Le cas d'un objet de forme symétrique a aussi été étudié. Et les résultats sont très satisfaisants. L'ensemble de ces simulations montre que le système converge pour de très importants déplacements. Quel que soit le traitement de l'image utilisé pour extraire les coordonnées des points, au-

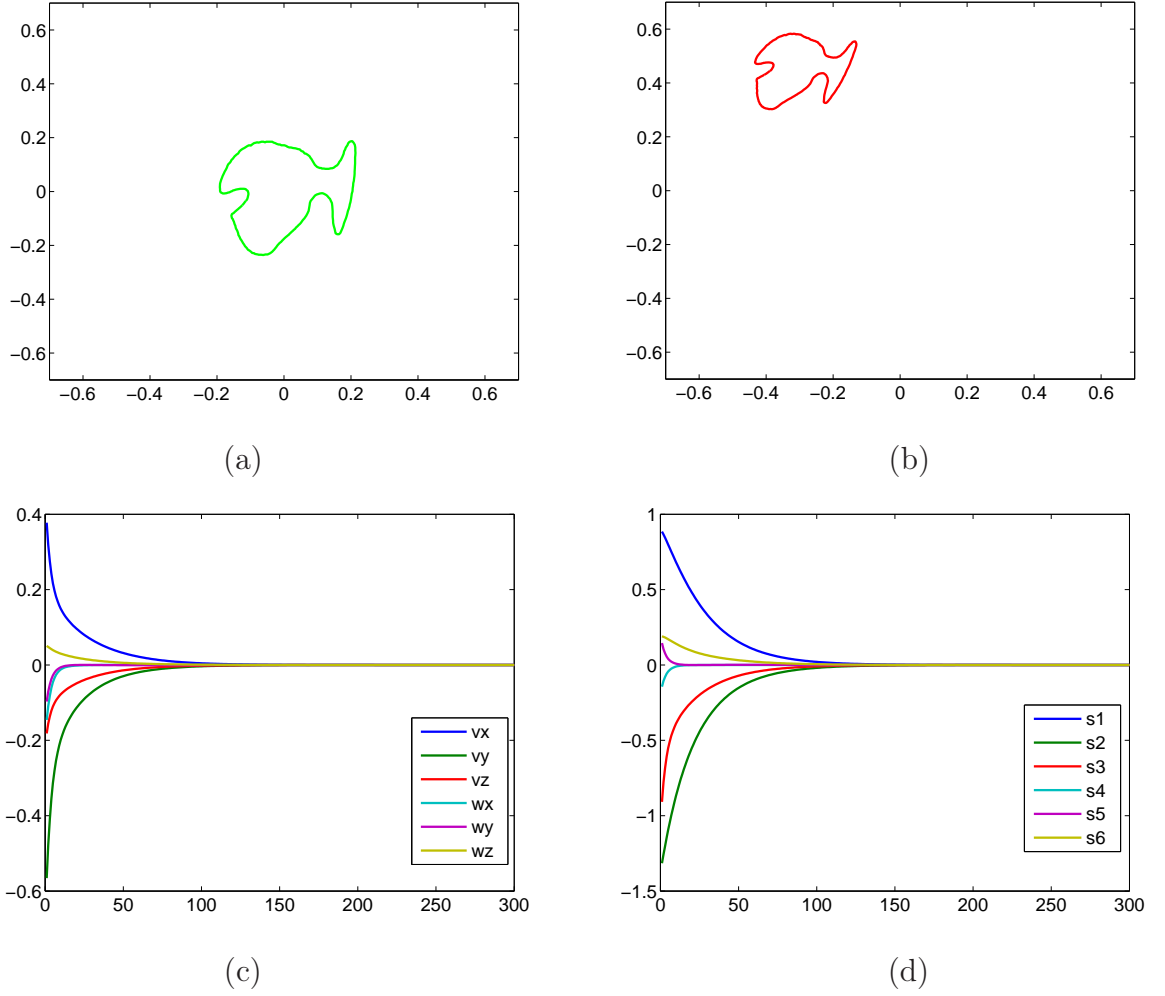


FIGURE 4.12 – Résultats de Simulation pour un mouvement générique avec comme paramètres de décalage " a_1 " et " a_2 " : image d'une baleine. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur " s ".

cun ne peut assurer de façon certaine de fournir des mesures exactes à chaque itération. L'intrusion d'éléments parasites dans l'image ou l'occultation partielle d'un des points de l'objet peut conduire l'algorithme de traitement d'images à fournir des mesures aberrantes. Si aucun mécanisme de sécurité n'est prévu dans la loi de commande, ces mesures sont utilisées par le contrôleur et il faut alors assurer que le système en boucle fermée ne devienne instable. De plus, comme toutes mesures issues d'un capteur, les informations visuelles sont entachées de bruit de mesure et le contrôleur doit assurer la convergence de la boucle de contrôle même en présence de ce bruit. La loi de commande proposée a été évaluée en présence de

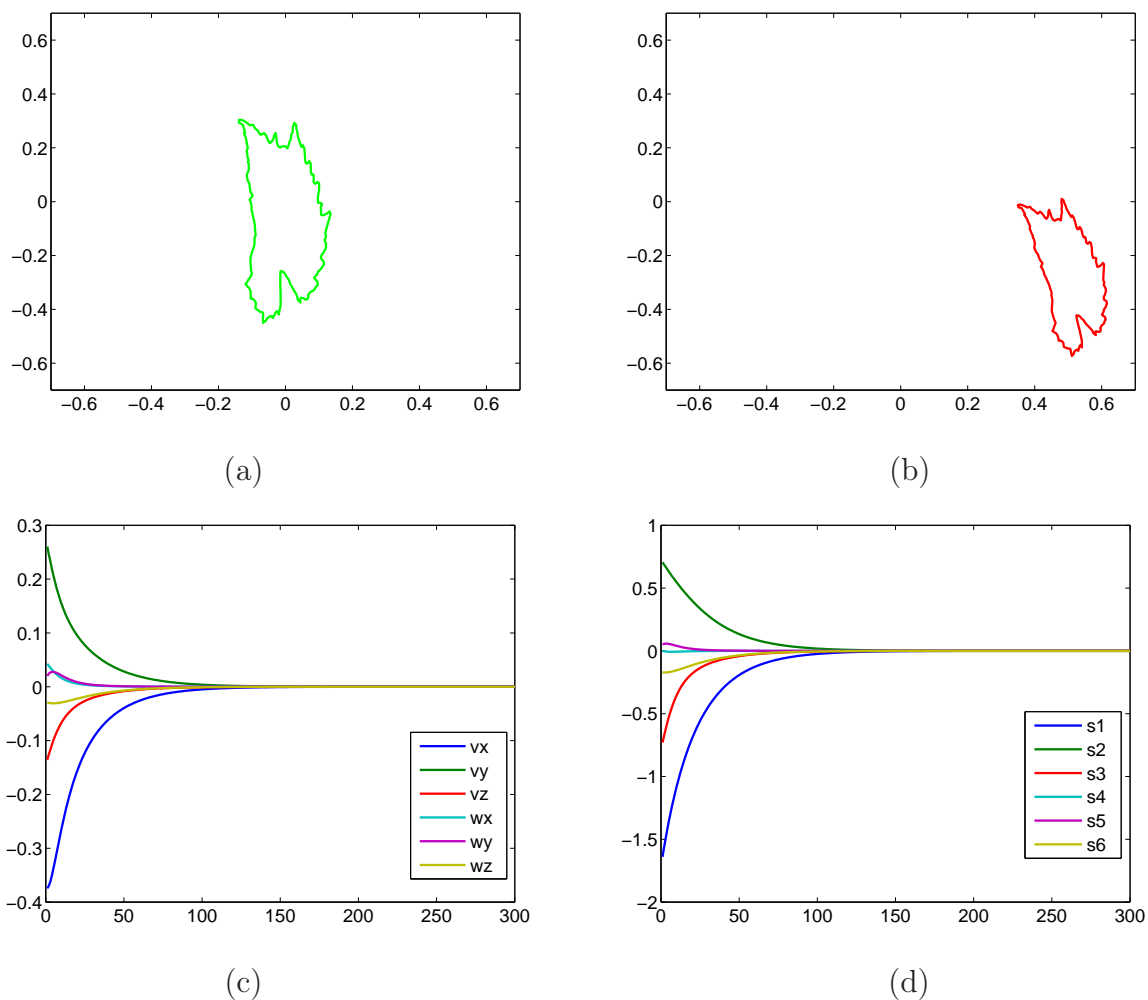


FIGURE 4.13 – Résultats de Simulation pour un mouvement générique avec comme paramètres de décalage " a_1 " et " a_2 " : image d'une feuille de vigne. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".

bruit de mesure sur les coordonnées des points dans l'image pour un mouvement complexe de l'objet. Les résultats obtenus montrent la convergence du système en présence d'un bruit de mesure uniforme centré d'amplitude 0.01.

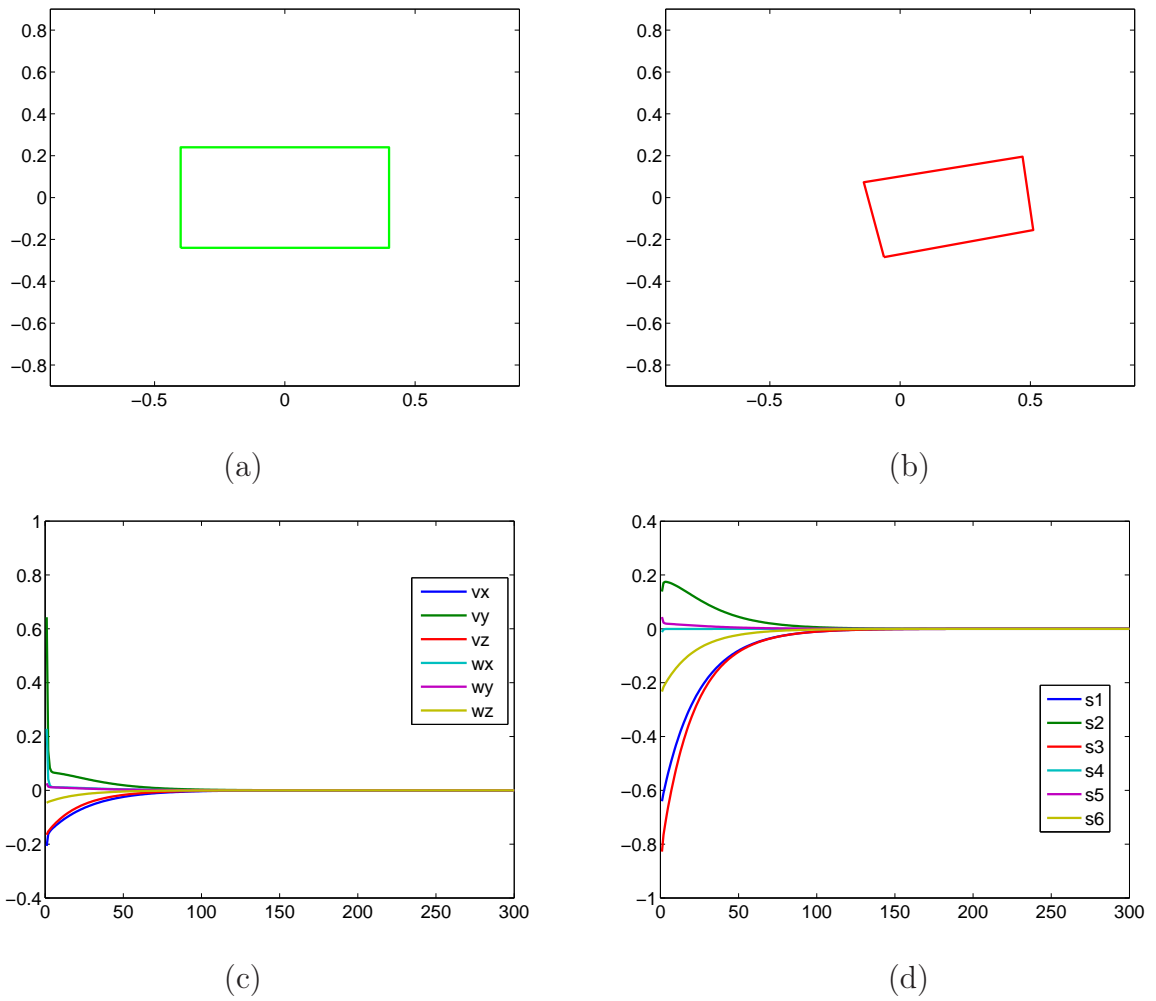


FIGURE 4.14 – Résultats de Simulation pour un mouvement générique avec comme paramètres de décalage " a_1 " et " a_2 " : image d'un rectangle. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur " \mathbf{s} ".

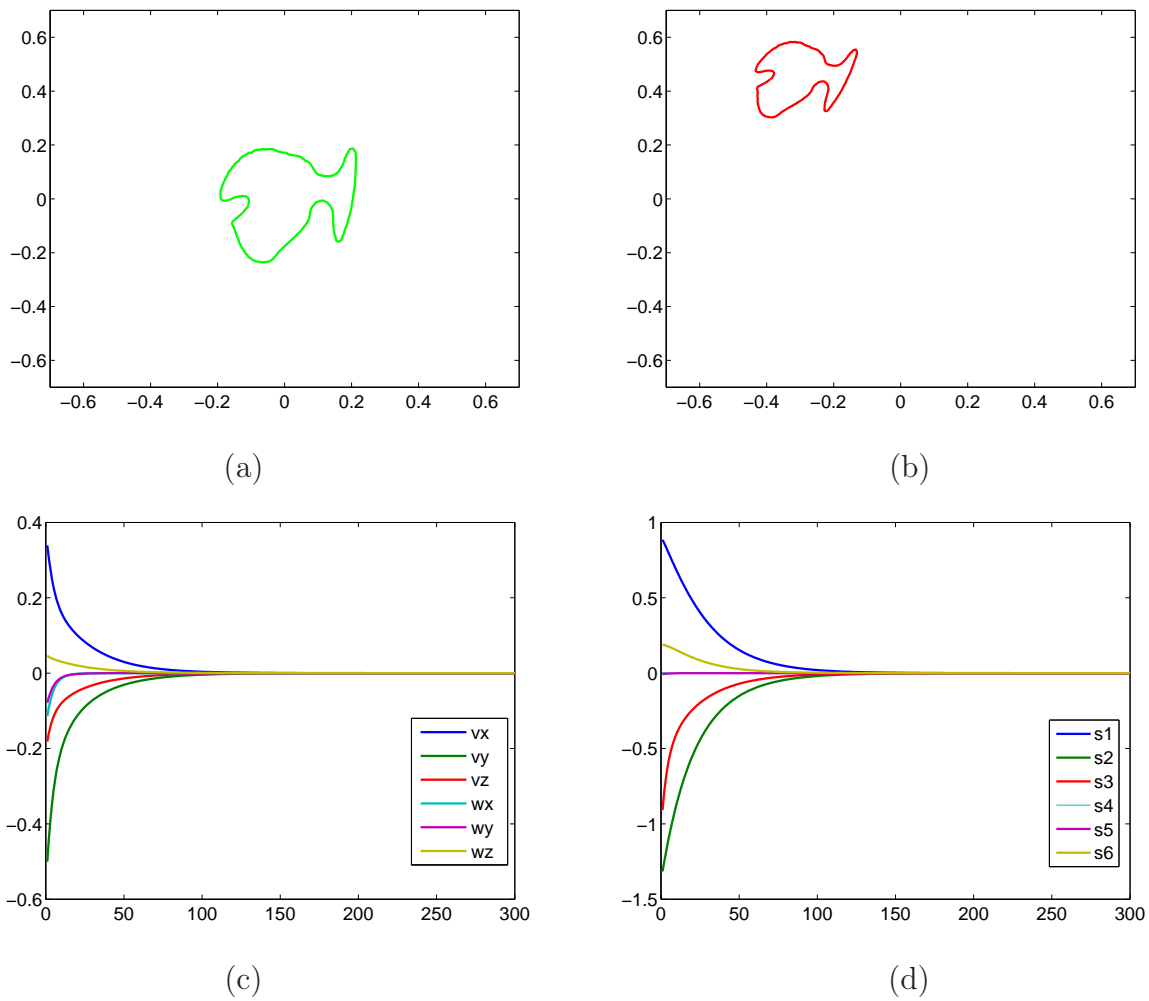


FIGURE 4.15 – Résultats de Simulation pour un mouvement générique avec comme invariant r_{s3} : image d'une baleine. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".

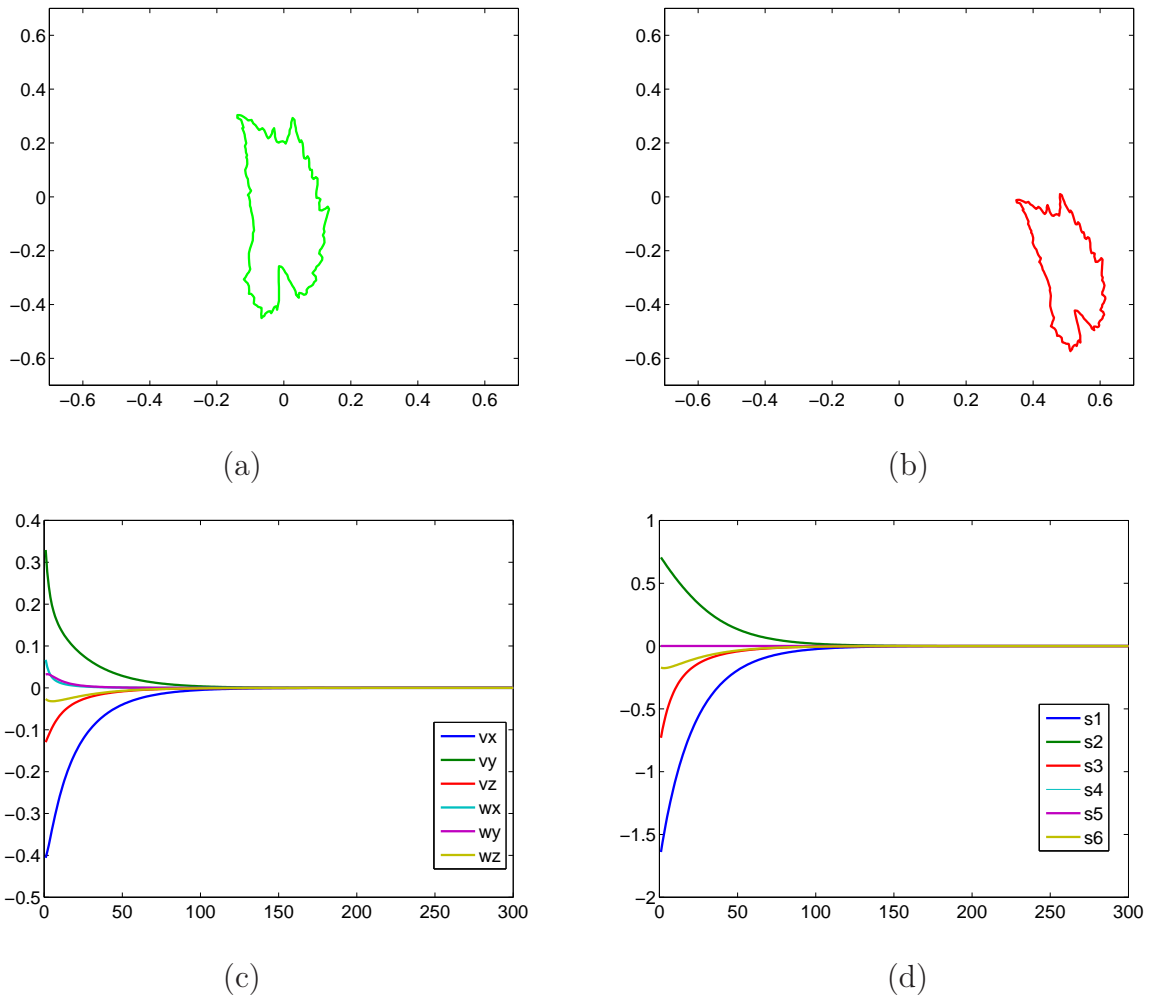


FIGURE 4.16 – Résultats de Simulation pour un mouvement générique avec comme invariant r_{s3} : image d'une feuille de vigne. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".

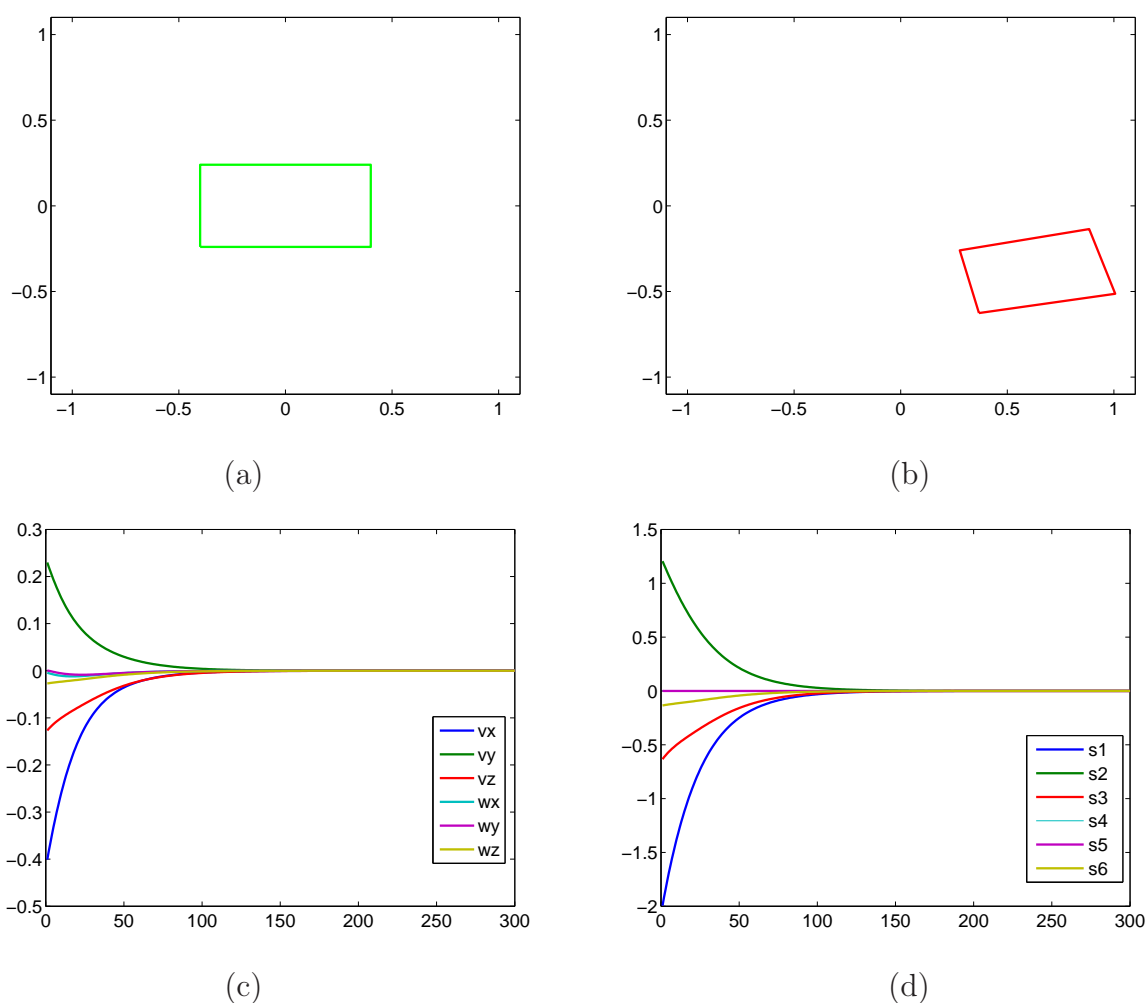


FIGURE 4.17 – Résultats de Simulation pour un mouvement générique avec comme invariant r_{s3} : image d'un rectangle. (a) image désirée, (b) image initiale, (c) vitesse de la caméra (en m/s et dg/s), (d) erreur sur "s".

Conclusion Générale et Perspectives

Conclusion Générale

Nous avons présenté dans ce manuscrit nos différentes contributions en asservissement visuel. La première contribution concerne une nouvelle méthode d'utilisation des moments "shifted" en lieu et place des moments géométriques afin de contrôler les mouvements de rotation de la caméra le long des axes w_x et w_y orthogonaux à l'axe optique. Le centre de gravité de l'objet est décalé en un autre point. Les coordonnées de ce nouveau centre sont choisies de telle sorte que toutes les propriétés d'invariance au translation, rotation et changement d'échelle soient préservées. La deuxième grande contribution de ce travail est le choix des informations visuelles qui ne dépend plus de la forme de l'objet. Les résultats obtenus montrent l'efficacité de notre approche. Enfin la dernière contribution est le choix des moments invariants d'ordre faible afin de minimiser l'impact du bruit dans la commande pour garantir la stabilité du système. Les simulations montrent que les informations visuelles proposées permettent d'assurer la convergence du système vers la position désirée en assurant le respect des contraintes liées à la mécanique du robot et celles induites par l'utilisation d'un capteur visuel. L'évaluation de la robustesse montre que la loi de commande est stable même en présence de perturbations. La forme analytique de la matrice d'interaction relative à l'utilisation des moments "shifted" sur des objets complexes définis par un contour fermé a été obtenue et six informations visuelles ont été sélectionnées afin de proposer un schéma de commande complètement découplé.

Choix des informations visuelles

Nous avons vu dans le Chapitre I que le choix des primitives dans le cadre de l'asservissement visuel [TFC09, NAH02] et des paramètres les décrivant [IO05, KI07], est primordial afin de réaliser la synthèse de lois de commande permettant d'obtenir les meilleures propriétés de linéarisation et de découplage. Dans les applications de commande de robot, la fonction de tâche définit un critère à optimiser qui s'apparente dans le contexte de l'optimisation non linéaire à la fonction de coût. Dans l'asservissement visuel 2D classique, cette fonction repré-

sente l'erreur entre la position des primitives visuelles observées et désirées dans l'image, transformées par la matrice de combinaison en un vecteur à 1 éléments correspondant aux 1 degrés de liberté à contrôler. Ainsi, dans les applications où le mouvement est défini dans l'espace, la dimension du vecteur calculé par la fonction de tâche est de 6×1 . En réalité, tous les modèles de projection utilisés dans l'élaboration de schémas de commande par vision utilisent des hypothèses qui permettent de simplifier les lois de commande obtenues.

Nouvelles Primitives

Dans le chapitre II, nous avons rappelé les notions sur la commande par asservissement visuel et en particulier la notion de matrice d'interaction. Par ailleurs, nous avons donné les formules analytiques de ces matrices dans le cas des moments géométriques. Nous avons présenté également les problèmes potentiels rencontrés en asservissement visuel. Pour remédier à ces problèmes, nous avons proposé six informations visuelles calculées à partir des moments géométriques d'une image d'un objet plan continu, dont le centre de gravité a été décalé. Ce choix d'informations visuelles a permis d'obtenir une matrice d'interaction creuse et de minimiser les non linéarités de celle-ci pour une position de la caméra où son plan image est parallèle à l'objet. Dans ce Chapitre nous avons présenté les contributions majeures de notre étude.

La modélisation obtenue au Chapitre II, nous a permis au troisième Chapitre de nous intéresser à l'étude de la stabilité. Dans ce chapitre relatif à l'étude de la stabilité, nous avons étudié la commande référencée capteur des robots manipulateurs en utilisant les mesures issues de capteurs embarqués. La problématique sous-jacente, liée au caractère très non linéaire des équations cinématiques des robots, est celle de la robustesse des propriétés de stabilité associées aux commandes considérées, vis-à-vis d'incertitudes sur le modèle du capteur. D'un point de vue théorique, nous avons étudié la stabilité des systèmes non linéaires et les résultats obtenus ont été insérés dans une boucle de commande des bras manipulateurs afin d'obtenir la loi de convergence et la condition de stabilité. Tous les résultats de simulations que nous avons obtenus dans le dernier chapitre ont confirmé ces propriétés. Nous avons ensuite proposé la détermination analytique des matrices d'interactions associées aux points \mathbf{P}_1 et \mathbf{P}_2 calculés à partir des nouveaux moments "shifted". Une condition nécessaire et suffisante de stabilité a été donnée.

Validations des résultats

Dans le dernier chapitre, nous avons validé les résultats théoriques sur le choix des primitives visuelles obtenus au chapitre 2 en utilisant les moments "shifted".

Les résultats de simulations ont été réalisés dans le cas idéal et dans un autre cas en présence du bruit. Cette commande est bien adaptée pour contrôler le déplacement du robot aussi bien dans le cas des mouvements simples que complexes et variés. Cette commande peut être appliquée à toute configuration d'objet représenté par un contour fermé. Le cas d'un objet de forme symétrique a aussi été étudié. Et les résultats sont très satisfaisants. L'ensemble de ces simulations montre que le système converge pour de très importants déplacements. Quel que soit le traitement de l'image utilisé pour extraire les coordonnées des points, aucun ne peut assurer de façon certaine de fournir des mesures exactes à chaque itération. L'intrusion d'éléments parasites dans l'image ou l'occultation partielle d'un des points de l'objet peut conduire l'algorithme de traitement d'images à fournir des mesures aberrantes. Si aucun mécanisme de sécurité n'est prévu dans la loi de commande, ces mesures sont utilisées par le contrôleur et il faut alors assurer que le système en boucle fermée ne devient pas instable. De plus, comme toutes mesures issues d'un capteur, les informations visuelles sont entachées de bruit de mesure et le contrôleur doit assurer la convergence de la boucle de contrôle même en présence de ce bruit. La loi de commande proposée a été évaluée en présence de bruit de mesure sur les coordonnées des points dans l'image pour un mouvement complexe de l'objet. Les résultats obtenus montrent la convergence du système en présence d'un bruit de mesure uniforme centré d'amplitude 0.01.

Perspectives

De nombreuses questions restent en suspens au termes de cette étude.

- d'un point de vue théorique, nous avons pu obtenir au chapitre III la forme analytique des matrices d'interactions relatives aux points \mathbf{P}_1 et \mathbf{P}_2 et une condition nécessaire et suffisante de stabilité pour que la matrice d'interaction relative aux informations visuelles choisies soit non singulière. Il serait intéressant de déterminer la courbe de singularité ;
- d'un point de vue pratique, il serait aussi intéressant de porter ces résultats de simulations sur une plate forme expérimentale équipée d'un robot à six degrés de liberté comme le robot cartésien AFMA.

Bibliographie

- [Agi77] G.J. Agin. Servoing with visual feedback. In In Proc. of 7th Int. Sympo On Industrial Robotics, Tokyo, Japan, October 1977.
- [BE92] F. Chaumette et P. Rives B. Espiau. A new approach to visual servoing in robotics. IEEE Transactions of Robot ics and Automation, 8(3) :313–326, Juin 1992.
- [Bey92] H.A. Beyer. Accurate calibration of ccd-cameras. In International Conference on Computer Vision and Pattern Recognition, 64 :96–101, 1992.
- [Bou94] B. Boufama. Reconstruction tridimensionnelle en vision par ordinateur : cas des caméras non étalonnées. PhD thesis, Institut Polytechnique de, Grenoble, décembre 1994.
- [BTG02] L. Cordesses B. Thuilot, P. Martinet and J. Gallice. Position based visual servoing : keeping the object in the field of vision. In Proc. IEEE International Conference on Robotics and Automation, pages 1624–1629, 2002.
- [CC00] C. Collewet and F. Chaumette. A contour approach for image-based control of objects with complex shape. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'00, volume 1, pages 751–756, Takamatsu, Japon, November 2000.
- [CC01a] A. Crétual and F. Chaumette. Application of motion-based visual servoing to target tracking. Int. Journal of Robotics Research, 20(11) :878–890, November 2001.
- [CC01b] A. Crétual and F. Chaumette. Visual servoing based on image motion. Int. Journal of Robotics Research, 20(11) :857–877, November 2001.
- [CC02] C. Collewet and F. Chaumette. Positioning a camera with respect to planar objects of unknown shape by coupling 2d visual servoing and 3d estimations. IEEE Trans. on Robotics and Automation, 18(3) :322–333, June 2002.

- [CH01] P. I. Corke and S. A. Hutchinson. new partitioned approach to image-based visual servo control. *IEEE Transaction on Robotics and Automation*, 17(4) :507–515, August 2001.
- [Cha90] F. Chaumette. La relation vision-commande : théorie et application à des tâches robotiques. PhD thesis, Université de Rennes 1, Mention informatique, Rennes, 1990.
- [Cha98] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G. Hager, and A.S. Morse, editors, *The Confluence of Vision and Control*, pages 66–78. LNCIS Series, No 237, Springer-Verlag, 1998.
- [Cha02] F. Chaumette. A first step toward visual servoing using image moments. In *IEEE/RSJ Int Conf on Intelligent Robots and Systems, IROS'02*, volume 1, pages 378–383, Lausanne, Switzerland, October 2002.
- [Cha03] F. Chaumette. Avancées récentes en asservissement visuel. In *Journées Nationales de la Recherche en Robotique, JNRR'03*, pages 103–108, Clermont-Ferrand, France, October 2003.
- [Cha04] F. Chaumette. Image moments : a general and useful set of features for visual servoing. *IEEE Trans. on Robotics*, 20(4) :713–723, August 2004.
- [CM09a] C. Collewet and E. Marchand. Colorimetry-based visual servoing. In *IEEE Int. Conf. on Intelligent Robots and Systems, IROS'09*, pages 5438–5443, St Louis, USA, October 2009.
- [CM09b] C. Collewet and E. Marchand. Photometry-based visual servoing using light reflexion models. In *IEEE Int. Conf. on Robotics and Automation, ICRA'09*, pages 701–706, Kobe, Japan, May 2009.
- [CM11] C. Collewet and E. Marchand. Photometric visual servoing. *IEEE Trans. on Robotics*, 27(4) :828–834, August 2011.
- [CS91] M. Le Borgne et B. Espiau C. Samson. *Robot control : the task function approach*. Clarendon Press, Oxford, United Kingdom 1991.
- [CTMC04] A.I. Comport, O. Tahri, E. Marchand, and F. Chaumette. Visual servoing with respect to complex objects. In *Int. Symp. on Robotics, ISR'04*, Paris, France, March 2004.
- [Deg98] K. Deguchi. Optimal motion control for image-based visual servoing by decoupling translation and rotation. In *Proc. IEEE/RSJ Int. Conf. of Intelligent Robots and Systems (IROS'98)*, pages 705–711, Victoria, Canada, 1998.

- [Deg00] Koichiro Deguchi. A direct interpretation of dynamic images with camera and object motions for vision guided robot control. *International Journal of Computer Vision*, pages 7–20, 2000.
- [DeM92] L. S. DeMenthon, D. F. et Davis. Model-based object pose in 25 lines of code. *Proc. European Conf. on Computer Vision (ECCV'92)*, 1992.
- [EBC04] S. Tarbouriech et J.M. Gomes Da Silva Jr E. B. Castelan, I. Queinnec. Lmi approach for i2-control of linear systems with saturating actuators. *6th IFAC-Symposium on Nonlinear Control systems (NOLCOS'2004)*, pages 287–292, September 2004.
- [Esp93] B. Espiau. Effects of camera calibration errors on visual servoing in robotics. In *Proc. Int. Symp. on Experimental Robotics (ISER'93)*, Kyoto, Japon, 1993.
- [Fau93] O. Faugeras. *Three-dimensional computer vision : a geometric viewpoint*. MIT Press , Cambridge, MA 1993.
- [Has98] T Hashimoto, K. et Noritsugu. Performance and sensitivity in visual servoing. *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'98)*, 2 :2321–2326, 1998.
- [Hu62] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2) :179–187, 1962.
- [HZ00] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000.
- [IO05] M. Iwatsuki and N. Okiyama. A new formulation of visual servoing based on cylindrical coordinate system. *IEEE Trans. on Robotics and Automation*, 21(2) :266–273, 2005.
- [JS03] Aubert G. Jehan S., Barlaud M. Dream2s : Deformable regions driven by an eulerian accurate minimization method for image and video segmentation. In *International Journal on Computer Vision* 53, pages 40–70, 2003.
- [KDS⁺07] Vinutha Kallem, Maneesh Dewan, John P. Swensen, Gregory D. Hager, and Noah J. Cowan. Kernel-based visual servoing. In *IROS*, pages 1975–1980, 2007.
- [KH96] H. Kimura K. Hashimoto, T. Ebine. Visual servoing with hand-eye manipulator- optimal control approach. *Proc. IEEE Int. Conf. on Robotics and Automation*, 12 :766–774, October 1996.
- [KI07] T. Komuro and M. Ishikawa. A moment-based 3d object tracking algorithm for high-speed vision. In *Proc. IEEE International Conference on Robotics and Automation*, pages 58–63, April 2007.

- [LD08] E. Laroche and J. Delavigne. Dynamical effects of vision-based position measurement. pages 7522–7527, In IFAC World Congress, 2008.
- [Low91] D. Lowe. Fitting parametrized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 5 :441–450, 1991.
- [LW87] A. Sanderson et C. Neuman L. Weiss. Dynamic sensor-based control of robots using visual feedback. *IEEE Journal of Robotics and Automation*, 3(5) :965–980, October 1987.
- [LWB96] C.W Hulls L.W Wilson and G.S Bell. Relative end-effector control using cartesian position based visual servoing. *IEEE Trans. on Robotics and Automation*, 12(5) :684–696, October 1996.
- [Mal98] E. Malis. Contributions à la modélisation à la commande en asservissement visuel. PhD thesis, Université de Rennes 1, Mention informatique, Rennes, 1998.
- [Mar07] E. Marchand. Control camera and light source positions using image gradient information. In *IEEE Int. Conf. on Robotics and Automation, ICRA’07*, pages 417–422, Roma, Italy, April 2007.
- [MC02a] E. Malis and F. Chaumette. Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. *IEEE Trans. on Robotics and Automation*, 18(2) :176–186, April 2002.
- [MC02b] E. Marchand and F. Chaumette. Virtual visual servoing : A framework for real-time augmented reality. In G. Drettakis and H.-P. Seidel, editors, *EUROGRAPHICS 2002 Conference Proceeding*, volume 21(3) of *Computer Graphics Forum*, pages 289–298, Saarebrücken, Germany, September 2002.
- [MC07] N. Mansard and F. Chaumette. Task sequencing for sensor-based control. *IEEE Trans. on Robotics*, 23(1) :60–72, February 2007.
- [MC10] E. Marchand and C. Collewet. Using image gradient as a visual feature for visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS’10*, Taipei, Taiwan, October 2010.
- [MCB99] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 d visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2) :238–250, April 1999.
- [MD89] J.T. Lapreste G. Rives M. Dhome, M. Richetin. Determination of the attitude of 3d objects from a single perspective image. . *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 11(12) :1265–1278, December 1989.
- [Meb10] R. Mebarki. Automatic guidance of robotized 2D ultrasound probes with visual servoing based on image moments. PhD thesis, Université de Rennes 1, Mention Traitement du Signal, March 2010.

- [Mic93] P Michel, H. et Rives. Singularities in the determination of the robot effector from the perspective view of 3 points. Technical Report 1850, INRIA, 1993.
- [MKC10] R. Mebarki, A. Krupa, and F. Chaumette. 2d ultrasound probe complete guidance by visual servoing using image moments. *IEEE Trans. on Robotics*, 26(2) :296–306, April 2010.
- [ML95] J-M. Lavest M. Li. Some aspects of zoom-lens camera calibration. Technical Report ISRN KTH/NA/P-95/03-SE, Laboratory, Royal Institute of Technology, Stockholm, Sweden, Février 1995.
- [Mor99] Szewczyk-J. Boudet S. et Pot J. Morel, G. Explicit incorporation of 2d constraints in vision based control of robot manipulators. In *Proc. IEEE Int. Symp. on Experimental Robotics (ISER'99)*, pages 99–108, Sidney, Australie, 1999.
- [NAH02] B. Espiau N. Andreff and R. Horaud. Visual servoing from lines. *The International Journal of Robotique Research*, 21(8) :679–700, 2002.
- [Nel95] P. Nelson, B. et Khosla. Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limits avoidance. In *Int. J. of Robotics Research*, volume 14(3), pages 255–269, 1995.
- [NNM96] S.K. Nayar, S.A. Nene, and H. Murase. Subspace Methods for Robot Vision. *IEEE Transactions on Robotics and Automation*, 12(5) :750–758, Oct 1996.
- [NPK83] P.K. Khosla N.P. Papanikolopoulos and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot : a combination of control and vision. *IEEE Transaction on Robotics and Automation*, 9(1), 1983.
- [Pap95] N.P. Papanikolopoulos. Selection of features and evaluation of visual measurements during robotic visual servoing tasks. *Journal of intelligent and robotics systems*, 1995.
- [PKA93] A. Timcenko et A. Michelman P. K. Allen, B. Yoshimi. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Transaction on Robotics and Automation*, 9(2) :152–165, Avril 1993.
- [PM99] J. Gallice P. Martinet. Position based visual servoing using a non-linear approach. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1 :531–536, October 1999.
- [PMG96] F. Berry P. Martinet and J. Gallice. Use of first derivative of geometric features in visual servoing. In *Proc. IEEE International Conference on Robotics and Automation*, 4 :3413–3419, 1996.

- [RH93] O. Monga R. Horaud. Vision par ordinateur, outils fondamentaux. Hermès , Paris, France, August 1993.
- [Rém98] S. Rémy. Etalonnage d'un système de vision embarqué. PhD thesis, Université Blaise Pascal, Clermont II, Juillet 1998.
- [Shi73] H. Shirai, Y. et Inoue. Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, 5(2) :99–106, 1973.
- [SJ07] D. Santosh and C. Jawahar. Visual servoing in non-rigid environments : A space-time approach. In *Proc. IEEE International Conference on Robotics and Automation*, pages 10–14, April 2007.
- [Tah04] O. Tahri. Application des moments à l'asservissement visuel et au calcul de pose. PhD thesis, Université de Rennes 1, mention informatique, Rennes, mar 2004.
- [TC88] C.H. Teh and R.T. Chin. On image analysis by the methods of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10 :496–513, 1988.
- [TC05a] O. Tahri and F. Chaumette. Complex objects pose estimation based on image moment invariants. In *IEEE Int. Conf. on Robotics and Automation, ICRA'05*, pages 438–443, Barcelona, Spain, April 2005.
- [TC05b] O. Tahri and F. Chaumette. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Trans. on Robotics*, 21(6) :1116–1127, December 2005.
- [Tea80] M. R. Teague. Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70 Issue 8 :920–930, 1980.
- [TFC09] R. Tatsambon Fomena and F. Chaumette. Improvements on visual servoing from spherical targets using a spherical projection model. *IEEE Trans. on Robotics*, 25(4) :874–886, August 2009.
- [TO00] Camillo J. Taylor and James P. Ostrowski. Robust vision-based pose control. *IEEE Trans. on Robotics and Automation*, pages 2734–2740, April 2000.
- [Tsa86] Roger Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'86)*, 3(4) :364–374, 1986.
- [Tsa87] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4) :323–344, 1987.
- [WW96] G.S Bell W.J Wilson, C.C Williams Hulls. Relative end-effector control using cartesian position-based visual servoing. . *IEEE Trans. on Robotics and Automation*, 12(5) :684–696, 1996.

- [YC89] A. Desrochers Y. Chen. Minimum-time control laws for robotic manipulators. Proc. of the 28th Conference on Decision and Control, pages 2494–2499, 1989.
- [Zha96] Z. Zhang. On the epipolar geometry between two images with lens distortion. International conference on Pattern Recognition, 1(1) :407–411, Août 1996.